(1)

A NUMERICAL SOLUTION OF THE TIME-DEPENDENT BOLTZMANN EQUATION

THESIS

Gaylord E. Seger, III
Captain, USAF

AFIT/GEP/ENP/89D-10

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
Wright-Patterson Air Force Base, Ohio

AFIT/GEP/ENP/89D-10

A NUMERICAL SOLUTION OF THE TIME-DEPENDENT BOLTZMANN EQUATION

THESIS

Gaylord E. Seger, III
Captain, USAF

AFIT/GEP/ENP/89D-10

AFIT/GEP/ENP/89D-10

A NUMERICAL SOLUTION OF THE TIME-DEPENDENT BOLTZMANN EQUATION

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

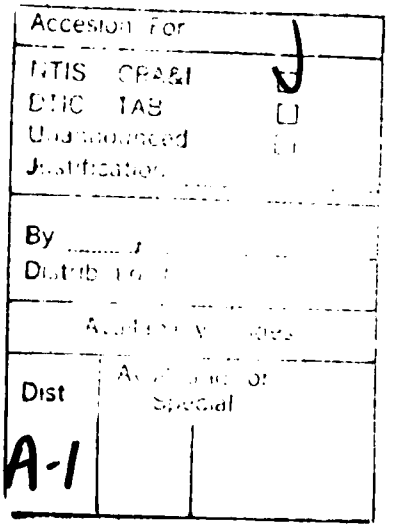
Requirements for the Degree of

Master of Science in Engineering Physics

Gaylord E. Seger, III, B. S.

Captain, USAF

October 1989



Approved for public release; distribution unlimited

Preface

The purpose of this thesis was to develop and test a computational model for simulating electron behavior in a gas by solving the time-dependent Boltzmann Equation. Immediate uses for this model are in providing needed data to succesfully model a gas-discharge laser or plasma-etching source.

Fairly extensive testing was performed on the MEGABOLTZ algorithm for all interactions and processes mentioned. Due to time constraints, I was unable to code a version of the program which supported variable energy binwidth, which I believe would improve program accuracy. Even so, the version of MEGABOLTZ which is referenced in this writeup tested out well against previously-reported data modeled by older programs. Also, due to time constraints, I was unable to incorporate a subroutine to handle attachment processes. While the program validations I performed for this project did not require modeling this process, I feel a robust version of MEGABOLTZ should know how to handle attachment processes should a problem involving them need to be calculated.

The assistance and contributions of other people were a great help to me in completing this thesis. I am deeply indebted to my faculty advisor, Dr. W. F. Bailey, for his guidance, support and almost limitless patience throughout my work. I am also indebted to Lt. Col. Jim Lupo, AFIT/ENP, for answering all those little "stupid FORTRAN/METALIB questions" I always seemed to have, and showing me a few UNIX tricks in the process which sped program development; and fellow classmate Capt. David Honey, for a different perspective on the same problem, friendly competition, and invaluable assistance in the early phases of program development in the form of previously-calculated data from his thesis. Thanks should also go to Kris Larsen, BLACKBIRD sysop, for providing an ear to gripe to whenever something went wrong on the computers (a fairly frequent occurrence during the quarter the bulk of this work was done); and Dr. Art Greene of Los Alamos National Labs, who gave up some of his vacation time this summer to give both Capt. Honey and I valuable guidance at the beginning of our projects. Finally, though I never met him

personally, I would like to acknowledge a debt of gratitude to Dr. Steven Rockwood, for it was his work over the years on this problem that provided the largest foundation which I built upon and the target I strove to reach.

Table of Contents

List of Figures

List of Tables

Table of Symbols Used In This Thesis

Symbol	Units	Description	Equation First Used
$\left(\frac{\delta f}{\delta t}\right)_{collisions}$		Collision term in terms of f	1
$\left(\frac{\delta n}{\delta t}\right)_{collisions}$		Collision term in terms of n	2
$\underline{1}$		Identity Matrix	11
$\langle\varepsilon\rangle$	eV	Average energy of electron EDF	18
a	$cm\ s^{-2}$	Electron acceleration	1
a_k	s^{-1}	Rate at which electrons gain energy through external field and elastic processes	3
$\bar{a}_k$	s^{-1}	Rate which electrons promoted from bin k due to electric field	13
A_{kj}	$cm^3\ s^{-1}$	Rate which electrons gaining energy from bin k de-excite electrons from bin j	10a
a_k^l	s^{-1}	Rate which electrons promoted from bin k due to electron-electron interactions	10a
b_k	s^{-1}	Rate at which electrons lose energy through external field and elastic processes	3
$\bar{b}_k$	s^{-1}	Rate which electrons demoted from bin k due to electric field	13
B_{kj}	$cm^3\ s^{-1}$	Rate which electrons losing energy from bin k excite electrons from bin j	10b
b_k^l	s^{-1}	Rate which electrons demoted from bin k due to electron-electron interactions	10b

$\underline{C}$	s^{-1}	Rate matrix for all electron-neutral interactions	8
$\Delta(sj)$	eV	Energy lost or gained due to inelastic interaction j of neutral species s	5
δ_{1j}		Delta function; causes secondary electrons due to ionization to appear in lowest energy bin.	
$\Delta\varepsilon$	eV	Bin width	4a
D_f	$cm^2\ s^{-1}$	Electron diffusion coefficient	17
E	$V\ cm^{-1}$	Electric field	2
ε	eV	Electron energy	2
$\varepsilon_{characteristic}$	eV	Characteristic electron energy	20
e	C	Electron charge	2
ε_k	eV	Energy in the middle of bin k	4a
f	$cm^{-3}\ eV^{-3/2}$	Electron distribution function; functional form $\exp[-\beta\varepsilon]$	1
$f_0(v)$	$cm^{-3}\ eV^{-3/2}$	Unperturbed reduced electron distribution function; first term of spherical-harmonic expansion of f(v)	A-3
f_k	$eV^{-1/2}$	Reduced normalized distribution; used to calculate electron diffusion coefficient.	18
$f_1(v)$	$cm^{-3}\ eV^{-3/2}$	Second term of spherical-harmonic expansion of f(v)	A-3
h	s	Time step	8
φ		Grouping of constants in external-field flux to speed up the derivation.	A-9
J_{ee}	$cm^{-3}\ eV^{-1}\ s^{-1}$	Flux of electrons in energy space due to electron-electron interactions	B-2

J_{el}	$cm^{-3}\ s^{-1}\ eV^{-1}$	Electron flux in energy space due to elastic collisions	2
J_f	$cm^{-3}\ s^{-1}\ eV^{-1}$	Electron flux in energy space due to external fields	2
k		Bin number along finite-differenced energy axis	3
k_b	$ev\ s^{-1}$	Botzmann's constant	4a
Λ		Ratio of electron mean free path to Debye length	B-1
λ_c		Largest eigenvalue of the $\underline{C}$ matrix	22
λ_t		Largest eigenvalue of the $\underline{T}$ matrix	22
μ	$cm^2\ s^{-1}\ eV^{-1}$	Electron mobility	20
m_e	kg	Electron mass	2
M_s	kg	Molecular weight of neutral species s	4a
m_{sj}		*Integerized offset along* energy axis; used for inelastic interaction j of neutral species s	5
N	cm^{-3}	Neutral number density	2
n	$cm^{-3}\ eV^{-1}$	Electron energy distribution function, functional form $\varepsilon^{1/2} \exp[-\beta\varepsilon]$	2
ν_k	s^{-1}	Electron-neutral elastic collision frequency	4a
$\bar{\nu}_k$	s^{-1}	Average electron-neutral elastic collision frequency	4a
n_o	cm^{-3}	Total electron number density	B-1
N_{sj}	cm^{-3}	Excited neutral number density for inelastic interaction j of neutral species s	6

Q	cm^2	Another way of expressing elastic collision cross-section	A-5
q_s		Molar fraction of neutral species s	4a
R^i_{sjk}	$cm^3\ s^{-1}$	Rate at which electrons lose energy m_{sj} from ionization interaction j of neutral species s	7
$\rho_k(t)$	$cm^{-3}\ eV^{-1}$	Intermediate vector representing right-hand side of Equation 11	
R^l_{sjk}	$cm^3\ s^{-1}$	Rate at which electrons gain energy m_{sj} from inelastic interaction j of neutral species s	6
R_{sjk}	$cm^3\ s^{-1}$	Rate at which electrons lose energy m_{sj} from inelastic interaction j of neutral species s	5
σ_{sjk}	cm^2	Cross-section of inelastic process j of neutral species s	5
σ_{sk}	cm^2	Elastic collision cross-section of neutral species s as a function of energy	4a
$\underline{T}[n_k(t)]$	s^{-1}	Electron-electron rate matrix	11
T_g	K	Gas temperature	4a
T_v	eV	Temperature of inelastic state	6
u	$cm^2\ s^{-2}$	Square of electron velocity	A-5
v	$cm\ s^{-1}$	Electron velocity	1
v_d	$cm\ s^{-1}$	Electron drift velocity	16
$\frac{\partial E_{el}}{\partial t}$	$eV\ s^{-1}$	Rate which electrons lose energy due to elastic collisions	14
$\frac{\partial E_g}{\partial t}$	$eV\ s^{-1}$	Rate which electrons gain energy from the electric field	13

$\frac{\partial E_{inel}}{\partial t}$	$eV\ s^{-1}$	Rate which electrons lose energy due to all inelastic processes	15
∇_r		Spatial gradient	1
∇_v		Velocity-space gradient	1

Abstract

Interest in gas discharge phenomena for laser, ion source, and plasma processing applications has generated needs for the solution of the time-dependent Boltzmann equation. An algorithm is developed and tested to compute the electron energy distribution function (EDF), incorporating elastic, inelastic, superelastic, ionization, and electron-electron collisions. A new finite-differencing approach which eliminates low-energy instabilities inherent in previous techniques is developed and tested. An implicit-explicit Euler approximation technique was used for the algorithm to transform the resulting nonlinear differential equation into a system of finite-differenced linear equations. The system is then solved using LU-decomposition for efficient matrix computation. The program developed using this algorithm, MEGABOLTZ, is first put through basic shakedown tests to verify the correctness of the algorithm. Next, the program calculates an EDF for nitrogen gas for electric field to neutral number density ratios (E/N) ranging from 5 - 40 Townsend. Distributions computed by MEGABOLTZ were compared to previously-reported data from other Boltzmann equation solvers and experiments, and are found to be in good agreement with them. Future modifications are suggested for MEGABOLTZ to improve robustness and accuracy. A short user's manual for MEGABOLTZ is also included.

A Numerical Solution of the Time-Dependent Boltzmann Equation

I. Introduction

The Boltzmann equation is an equation of continuity in phase space, describing the flow of particles through phase space as a function of time. Solutions yield the time-dependent distribution function, which then establishes values of excitation and ionization rates. These rates are used for modeling gas-discharge and excimer lasers, lamps, ion sources, RF discharges, electron beams, afterglow, low and high-density plasmas, or any other system in which it is necessary to understand electron behavior.

Under the physical conditions of interest, collisional processes are not only numerous, but exhibit strong and significant structure as a function of energy. This structure is typically non-analytic in form, precluding an analytic solution to the Boltzmann Equation. In addition, the equation becomes non-linear in cases of high fractional ionization due to the importance of electron-electron interactions in these conditions. Thus, numerical methods must be used to obtain a time-dependent distribution function.

A numerical solution to the equation should have the following design attributes: speed, accuracy, minimal storage requirements, and reasonably simple input/output (I/O).

Starting with previous work on modeling the Boltzmann equation, I will establish the equations necessary for solving the time-dependent Boltzmann transport equation, and then cast them in a form suitable for numerical solution. I will then discuss the program structure of MEGABOLTZ, a program which solves the Boltzmann equation. Finally, I will discuss validation procedures and compare calculated solutions with both previous work and experimental data

II. Historical Development

Holstein -- The Basics

In 1946, Holstein developed the theory for solving the time-dependent Boltzmann Equation in six-dimensional velocity-position space to describe the behavior of a plasma within an electric field (8:367-384):

$$\frac{\partial f}{\partial t} + (\mathbf{v}\cdot\nabla_r)f + (\mathbf{a}\cdot\nabla_v)f = \left(\frac{\delta f}{\delta t}\right)_{collisions} \qquad (1)$$

where **v** is the electron velocity vector, **a** is the electron acceleration vector, and f is the electron velocity distribution function. Holstein started his analysis by assuming a spherical distribution of velocities, then expanded them in a spherical harmonic series to account for asymmetries arising from collisions and the imposed electric field, retaining only the first two terms. This work was developed primarily for treating AC discharges. He also showed how to generalize the solution for the case of DC external fields, verifying functional forms derived by Druyvesteyn in 1930 (5).

Rockwood and the Various Incarnations of NOMAD

The NOMAD code was first developed in 1973 for gas-discharge laser modeling at the Air Force Weapons Laboratory (AFWL) (15:373-390). In later years, it was adapted at Los Alamos National Labs (LANL) and various other institutions for other type of work. Its primary use was for calculating the electron energy distribution function given certain initial parameters such as gas mix and density, temperature, and electric field. From this, transport coefficients and pumping rates for gas-discharge lasers were calculated (15:373-390). The code was also used, with some modifications, to calculate laser-induced gas breakdown, to determine elastic and inelastic electron-impact cross-sections from experimental electron transport data, to calculate electron cooling in planetary atmospheres, and even determine energy deposition in intense neutron

sources. Most papers dealing with numerical solutions to the Boltzmann Equation since the early 1970s either reference this work or use the formalism it develops.

Variations on a Theme

Although the majority of this work was based on the last two sources cited, three other sources played a role in the development of this thesis. In each of these articles, a solution to the Botzmann Equation was required for their investigations. Sometimes, different (and revealing) approaches to solving the equation were used.

In 1975, Long, Bailey, and Garscadden reported a simplified reformulation of the time-independent Boltzmann Equation to investigate various gas discharges (9:471-475). This work is mentioned because the formulation presented provides an important intermediate step in getting from Holstein's work to Rockwood's work. The program developed was used to calculate electron drift velocities in various gas mixtures.

In 1976, Elliott and Greene investigated a variation of Rockwood's code and formalism for excimer laser modeling (6:2946 - 2953). They started by assuming no external fields, and modeled the electron beam as a source function. They also fixed a possible instability in Rockwood's formulas for momentum transfer rates by making a slight change in his original finite-differencing scheme. The program they developed used an implicit treatment of both electron-neutral and electron-electron interactions, and an explicit treatment of time-dependent recombination in the gas mix.

In a series of papers from 1981 to 1985, a team led by Bretagne developed a computer code for solving the time-dependent Boltzmann Equation utilizing variable energy stepsize and either pre-programmed differential equation solvers or a system of finite-differenced equations (2:2205, 3:811). The primary use of their program was to calculate various types of gas discharges for ion sources.

III. Theory

The Basic Boltzmann Equation

I will start by deriving the form of the Boltzmann equation which will be used in the numerical algorithm. The formalism which will be used in this derivation comes primarily from Rockwood, with some assistance from Holstein and Long, Bailey, and Garscadden (LBG). A full derivation is included in Appendix A.

The basic time-dependent Boltzmann equation for electrons, as shown before, is:

$$\frac{\partial f}{\partial t} + (\mathbf{v}\cdot\nabla_r)f + (\mathbf{a}\cdot\nabla_v)f = \left(\frac{\delta f}{\delta t}\right)_{\text{collisions}} \qquad (2)$$

This represents an equation of continuity for a particle flux in phase space. The equation is solved for f, which represents a distribution of an ensemble of particles in both coordinate and velocity space. It is typically applied to what is called a "Lorentzian gas", which is a system where light particles (such as electrons) are colliding with much heavier particles (such as neutral gas molecules). In situations where the restriction on charged particle mass is removed, such as in ion-neutral or electron-electron interactions, the equation must be cast in an integral form (1:404). In this investigation, which includes electron-electron, Proctor's treatment of Rosenbluth's integral form was used (15:2356).

Using a two-term spherical harmonic expansion in velocity space, and recasting the distribution function in terms of the electron fluxes in energy space, equation 1 becomes:

$$\frac{\partial f}{\partial t} = -\frac{\partial J_f}{\partial \mathcal{E}} - \frac{\partial J_{el}}{\partial \mathcal{E}} + \left(\frac{\delta f}{\delta t}\right)_{\text{other}} \qquad (3)$$

where

$$J_{el} = -\frac{2m_e}{M} \nu \varepsilon^{3/2} f_0 \qquad \text{and} \qquad J_f = \frac{2Ne^2(E/N)^2}{3(2m_e)^{1/2}} \frac{1}{\varepsilon^{1/2}} \left(\frac{\varepsilon}{Q} \frac{\partial f_0}{\partial \varepsilon} \right)$$

J_{el} and J_f represent the continuum flow of electrons in energy space due to elastic collisions and external electric fields respectively, while f_0 represents the first term of the spherical-harmonic expansion of f(v) and $\left(\frac{\delta f_0}{\delta t}\right)$ represents discrete jumps in electron energy due to various collisional processes. Finite-differencing equation (3) over K cells of width $\Delta\varepsilon$ on the energy axis results in:

$$\frac{\partial n_k}{\partial t} = a_{k-1} n_{k-1} - (a_k + b_k) n_k + b_{k+1} n_{k+1} + [\text{inelastic}] + [\text{superelastic}] + [\text{ion}] \qquad (4)$$

Instead of solving for f, the distribution of particles in velocity space, equation (4) is solved in terms of n, the electron energy distribution function. $n(\varepsilon,t)$ represents the number density of electrons with energies from ε to $\varepsilon+\Delta\varepsilon$. I will spend the next sections explaining the other variables on the right-hand side of equation (4).

Momentum Transfer and External Field

I will start my detailed explanation with the coefficients a_k and b_k. They express the continuum flow of electrons resulting from both the applied electric field and momentum-transfer collisions, and can be interpreted as rates in which electrons are "promoted" or "demoted", respectively, in energy. They are determined by:

$$a_k = \frac{2Ne^2}{3m_e}\left(\frac{E/N}{\Delta\varepsilon}\right)^2\left[\left(\frac{\varepsilon_k}{\varepsilon_k^-}\right)^{1/2}\frac{\varepsilon_k}{\nu_k}\right] + \frac{\bar{\nu}_k}{4\Delta\varepsilon}\left[k_b T_g\left(1 + \frac{4\varepsilon_k}{\Delta\varepsilon}\right) - 2\varepsilon_k\right] \quad (5a)$$

$$b_{k+1} = \frac{2Ne^2}{3m_e}\left(\frac{E/N}{\Delta\varepsilon}\right)^2\left[\left(\frac{\varepsilon_k}{\varepsilon_{k+1}^-}\right)^{1/2}\frac{\varepsilon_k}{\nu_k}\right] + \frac{\bar{\nu}_k}{4\Delta\varepsilon}\left[k_b T_g\left(\frac{4\varepsilon_k}{\Delta\varepsilon} - 1\right) + 2\varepsilon_k\right] \quad (5b)$$

where

$$\nu_k = N(2\varepsilon_k/m_e)^{1/2}\sum_{sj} q_s\sigma_{sk} \qquad \bar{\nu}_k = 2m_e N(2\varepsilon_k/m_e)^{1/2}\sum_{sj}\frac{q_s\sigma_{sk}}{M_s}$$

$$\varepsilon_k^- = \varepsilon_k - \frac{\Delta\varepsilon}{2}$$

Inelastic

Having examined the continuum flow of electrons, I will now look at the remaining terms of equation (3). These terms describe the flow of electrons arising from various discontinuous electron-neutral interactions. I will begin this discussion with the term "[inelastic]" in equation (4). It describes electrons that undergo inelastic collisions with neutrals, losing energy $\Delta(sj)$ in the process (14:2349).

$$[\text{inelastic}] = \sum_{sj} N_s\left(R_{sjk+m(sj)}n_{k+m(sj)} - R_{sjk}n_k\right) \quad (6)$$

where

$$R_{sjk} = \sigma_{sjk}(2\varepsilon_k/m_e)^{1/2} \qquad m(sj) = \frac{\Delta(sj)}{\Delta\varepsilon}$$

R_{sjk} represents the rate in $cm^3\ s^{-1}$ in which electrons of energy $k\Delta\varepsilon$ collide with molecules of species s, losing energy $\Delta(sj)$ in the process. The energy lost by the electron excites state j in the molecule.

Superelastic

Next, the superelastic term describes electrons which undergo superelastic collisions with excited neutrals, gaining energy $\Delta(sj)$ in the process (14:2349).

$$\text{superelastic} = \sum_{sj} N_{sj} \left(R^{l}_{sjk-m(sj)} n_{k-m(sj)} - R_{sjk} n_k\right) \qquad (7)$$

where

$$R^{l}_{sjk} = \left(\frac{\varepsilon+\Delta(sj)}{\Delta\varepsilon}\right)\sigma_{sjk+m(sj)}(2\varepsilon_{k+m(sj)}/m_e)^{1/2}$$

$$m(sj) = \frac{\Delta(sj)}{\Delta\varepsilon} \qquad N_{sj} = N_s \exp[-\Delta(sj)/T_v]$$

N_{sj} represents the density of molecules of species s in excited state j in the gas mix, and T_v represents the vibrational temperature of the molecule (4:371).

Ionization

Finally, I will briefly discuss the ionization term. This term of equation (5) describes gas ionization by electron impact. An electron hits a neutral and loses energy $\Delta(sj)$ in the process. The new electrons generated by this impact are assumed to appear in the lowest energy bin (14:2349).

$$[\text{ion}] = \sum_{sj}\left[N_s\left(R^{i}_{sjk+m(sj)} n_{k+m(sj)} - R^{i}_{sjk} n_k\right) + \delta_{1j} \sum_{\Delta(sj)}^{K} R^{i}_{sjm} n_m\right] \qquad (8)$$

where

$$R_{sjk} = \sigma_{sjk}(2\mathcal{E}_k/m_e)^{1/2} \qquad m(sj) = \frac{\Delta(sj)}{\Delta\mathcal{E}}$$

Calculational Form

Equation (4) can be represented by the following system of linear differential equations (14:2349):

$$\frac{\partial n_k}{\partial t} = \underline{C}\, n_k(t) \tag{9}$$

The $\underline{C}$ matrix is a square matrix representation of all the rates (inelastic, elastic, etc.) calculated above. Recasting the time derivative using Euler's method, then grouping like terms of $n_k(t)$, we finally get the following linear system of equations:

$$\left[\underline{1} - \underline{C}h\right] n_k(t+h) = n_k(t) \tag{10}$$

Knowing the collision rates and the initial form of n(t), equation (10) can be solved for $n_k(t+h)$, then iterated in time until it converges upon a solution.

Boundary Conditions

One boundary condition applies to the equations I have shown: conservation of number density. In cases where gas ionization is unimportant, the number density of electrons should not change through the course of the calculation.This is assured by setting any electron-neutral collision rate which would cause an electron to have either a negative energy or an energy greater than the maximum energy being investigated equal to zero. An examination of Figures 1 and 2 is instructive at this point. Figure 1 is a pictoral representation of the elastic collision rates a_k and b_k, and Figure 2 is similar except that it shows the relation between the inelastic and superelastic rates. In both cases, for a steady-state solution ($\frac{\partial n_k}{\partial t} = 0$), the flux of electrons into cell k must equal the flux of electrons leaving cell k.

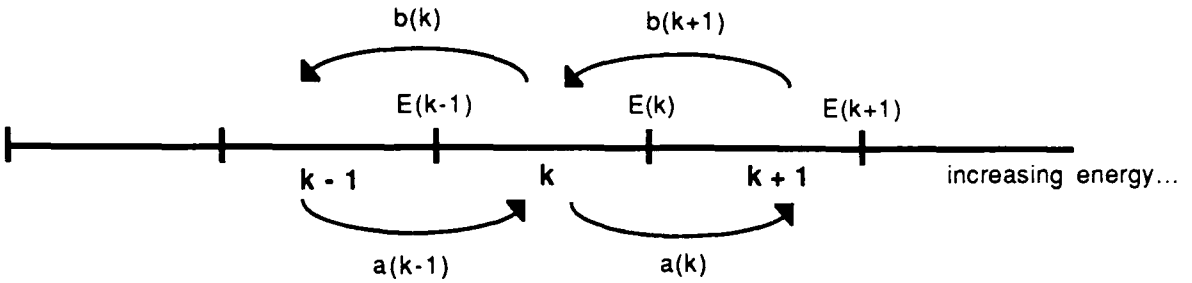


Figure 1: Diagram of Electron Fluxes in Energy Space for Momentum Transfer and External-Field Interactions.

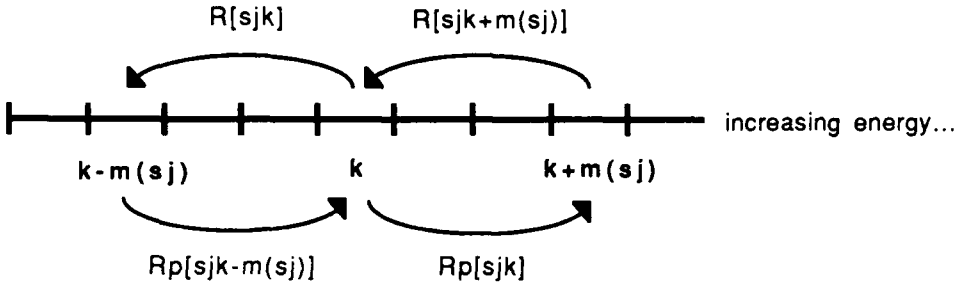


Figure 2: Diagram of Electron Fluxes in Energy Space for Inelastic and Superelastic Interactions.

Electron-Electron Collisions

A more complete treatment of electron-electron interactions, including a derivation of the terms used in this section, is included as Appendix B.

At very high electron number densities, such as for low E/N and high fractional ionization, electron-electron collisions dominate the shaping of the EDF. Instead of investing their energy in the neutral gas particles, the electrons now primarily redistribute the energy among themselves. To account for electron-electron interactions, the elastic collision rate vectors are modified as follows:

$$a_k^l = a_k + \sum_j A_{kj} n_j \qquad (11a)$$

$$b_{k+1}^l = b_{k+1} + \sum_j B_{k+1,j} n_j \qquad (11b)$$

where all variables follow the formalism originally developed by Proctor and Canavan, and reported by Rockwood (14:2349). The rates a_k and b_{k+1} are calculated in equations 6a and 6b above, and A_{kj} is computed according to equation (B-3) in Appendix B. B_{kj} is set equal to A_{jk}, as required by energy conservation. This is explained in detail in Appendix B. A_{kj} has units of (cm^3/sec), and represents the rate at which electrons of energy $\mathcal{E}_k$ are excited to energy $\mathcal{E}_{k+1}$ by interacting with electrons of energy $\mathcal{E}_j$, which drop to energy $\mathcal{E}_{j-1}$ in the process. These coefficient matrices are carefully constructed so to conserve particles and energy, but their use renders the system of equations previously developed non-linear by virtue of the fact they are now dependent on the value of $n_k(t)$ at each timestep. This non-linearity affects the way equation (10) is written(14:2350):

$$\left[\underline{1} - \underline{C}h\right] n_k(t+h) = \left[\underline{1} + \underline{I}[n_k(t)]h\right] n_k(t) \qquad (12)$$

This equation is an implicit-explicit differencing in time. The $\underline{I}$ matrix is composed of all the continuum flow rates due to electron-electron collisions. Since this matrix and $n_k(t)$ are known, we can perform straightforward matrix-vector multiplication on that side to get an intermediate vector $\rho_k(t)$. $\rho_k(t)$ is then used in place of $n_k(t)$ to solve for $n_k(t+h)$.

Boundary Conditions for Electron-Electron Interactions

Particle conservation is assured upon construction of the A_{kj} and B_{kj} matrices by doing the following (14:2357):

$$0 = A_{j1} = B_{jK} = A_{Kj} = B_{1j} \qquad (j=1,k) \qquad (13)$$

This insures that electrons will not be promoted or demoted off the energy axis, nor appear by a similar mechanism. Energy is conserved by setting B_{kj} equal to A_{jk}, as mentioned previously. A further boundary condition is imposed upon electron-electron interactions, in that a steady-state distribution of electrons only should form a Maxwellian distribution in energy space. This condition is insured by constructing the A_{jk} matrix in a certain way, using formulas developed in Appendix B.

IV. Computational Method

Program Structure

Now that the numerical method of calculating the Boltzmann equation has been developed, it is time to write a computer program based on it. MEGABOLTZ is the result. Given user input describing the physical situation of interest, MEGABOLTZ will calculate an electron energy distribution function (EDF) for that situation. With the assistance of Figure 3 and Table 1, I will now briefly describe the program's operation.

MEGABOLTZ first opens an input file named "input.com" in the subroutine GETSTUPH. After data input, the energy axis is initialized in ENINIT, electron-neutral collisional cross-sections are loaded for each gas specified by the user in LOOKUP, and the cross-sections linearly-interpolated to correspond with the generated energy axis in LINTERP. Once all this is done, INITIALIZE will call either MAXWELL or DRUYVESTEYN to generate the appropriate EDF for the initial guess.

Next, subroutine COEFF calls ENCOLL, INEL, SUPEREL, and ION to generate the $[\underline{1} - \underline{C}h]$ matrix, which consists of all external field, momentum transfer, inelastic, superelastic, and ionization-type electron-neutral collisions. This matrix only needs to be calculated once whenever the program is run. This means that the LU-decomposition of the matrix also can be calculated once, which is performed by either DLFTRG (IMSL library v. 10.0) or LUDATF (IMSL library v. [illegible]).

The electron-electron interaction matrix A_{kj} need only be calculated once in the course of the program, which is done by the subroutine EECOLL. With this step done, time iteration begins. The $\underline{\Pi}n(t)]$ matrix is calculated by SOLVR. This subroutine also calculates $[\underline{1} + \underline{I}h]\cdot n(t)$, which is simply the right-hand side of equation (12). Even though this routine has to be performed every iteration, SOLVR executes quickly by virtue of the fact that $\underline{\Pi}n(t)]$ is tridiagonal, minimizing the number of operations performed. Knowing the right-hand side of equation (12),

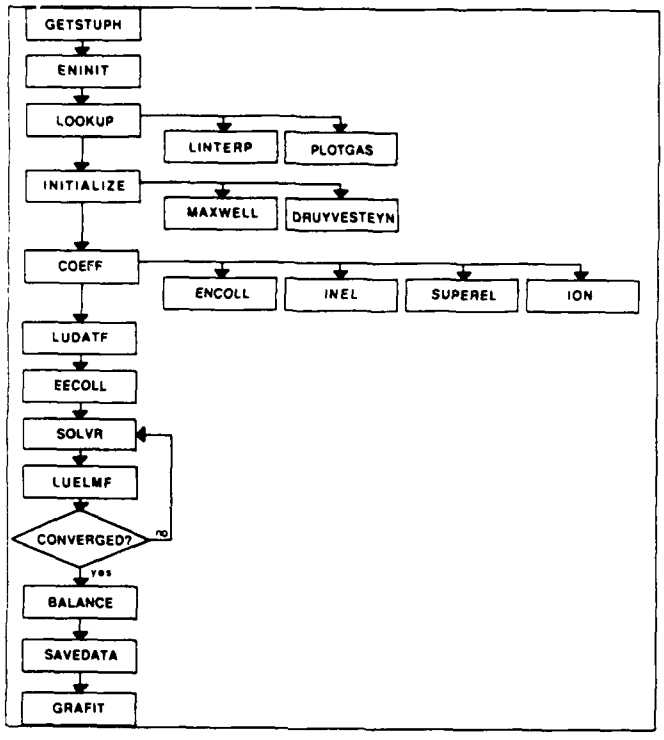


Figure 3: Program Flowchart for MEGABOLTZ.

Table 1:
Listing of Program Units, Subroutines, and Functions for MEGABOLTZ

Unit	What It Does
BOLTZMANN	Main program unit, calls all other subroutines and functions
GETSTUPH	Reads in data on E/N, gas mix temperature and pressure, electron number density, choice of initial guess function, mesh parameters for finite-differencing, and names for the save and plot files.
ENINIT	Initializes the energy axis based on user input.
LOOKUP	Reads in the gas mix and opens resident library files based on this input. Calls LINTERP and PLOTGAS
LINTERP	Linearly-interpolates library data on cross-sections to fit the calculated energy axis.
PLOTGAS	Plot interpolated cross-sections for each gas depending on if the user asks for it
INITIALIZE	Determines most prevalent gas in mix, pulls the first elastic cross-section, then calls either MAXWELL or DRUYVESTEYN depending on the user's choice of initial guess.
MAXWELL	Calculates a Maxwellian distribution for the initial guess based on input parameters.
DRUYVESTEYN	Calculates a Druyvesteyn distribution for the initial guess based on input parameters
COEFF	Loads electron-neutral collision matrix $\underline{C}$ by calling ENCOLL, INEL, SUPEREL, and ION, then calculates $\| \underline{1} - \underline{C}h \|$
ENCOLL	Calculates elastic collision rates

Unit	What It Does
INEL	Calculates inelastic collision rates
SUPEREL	Calculates superelastic collision rates
ION	Calculates ionization rates
EECOLL	Calculates A_{kj} matrix for electron-electron collisions
LUDATF DLFTRG	IMSL v. 9.2/10.0 routines to LU-decompose the $\| \underline{1} - \underline{C}h \|$ matrix.
SOLVR	Calculates the explicit (right-hand) side of the algorithm, given A_{kj} and n_k as input
LUELMF DLFSRG	IMSL v. 9.2/10.0 routine to backsolve implicit (left-hand) side of algorithm for $n_k(t+h)$ given the previously-decomposed $\| \underline{1} - \underline{C}h \|$ matrix and the previously-calculated right-hand side (from SOLVR)
LINFNORM	Function which finds maximum error between the previous and present iterations of n(t), used for convergence checking.
BALANCE	Calculates drift velocity, average energy, diffusion coefficient, and energy balance
SAVEDATA	Writes input data, initial guess, final calculated distribution, energy balancing, and calculated parameters to save file
GRAFIT	Generates plot file comparing initial guess to final calculated distribution using routines from the METALIB FORTRAN library

and possessing a previously-decomposed matrix from earlier program steps, n(t+h) is now calculated by the IMSL library routine DLFSRG (v. 10.0) or LUELMF (v. 9.2). Then, both n(t) and n(t+h) are used by the function LINFNORM, which returns the largest difference between the two. If this value is less that a pre-set tolerance (set to 10^{-4}), or if the maximum number of design iterations is exceeded (5000), time iteration is stopped. Otherwise, MEGABOLTZ loops back to SOLVR and continues to execute.

Once an EDF has been calculated, it is passed to the subroutine BALANCE for calculation of transport properties and an energy balance check. The final distribution, input parameters, calculated transport properties, and energy balance information is then written to a text file by SAVEDATA. Finally, a METALIB graph file of the reduced EDF is generated by the subroutine GRAFIT.

Checks During Calculation

Two methods are used to check how well the EDF calculation is proceeding. During computation, number density in each bin is checked while data is being compiled for convergence checking each iteration. If a negative number density occurs anywhere in the calculated distribution during the accumulation of these values, the calculation (for whatever reason) is going unstable and will likely give one bad values for drift velocity and diffusion coefficient. This type of error is trapped by the program at this point, warning the user while stoping calculation.

Conservation of energy is checked by working energy balance (14:2350, 15:380). The energy lost to all collisional processes should equal the energy gained by the electric field to a factor of 10^{-4} or better. This will result from a finite-differencing mesh which both extends out far enough in energy space to "capture" the majority of electrons, and is fine enough to resolve any

structure within the calculated distribution. Following Rockwood's formalism, the rate at which electrons gain energy from the external field is given by (14:2350):

$$\frac{\partial E_g}{\partial t} = \sum_k (\bar{a_k} - \bar{b_k})\, n_k \Delta\varepsilon \tag{14}$$

where

$$\bar{a_k} = \frac{2Ne^2}{3m_e}\left(\frac{E/N}{\Delta\varepsilon}\right)^2\left[\left(\frac{\varepsilon_k}{\bar{\varepsilon_k}}\right)^{1/2}\frac{\varepsilon_k}{\nu_k}\right] \tag{15a}$$

$$\bar{b_{k+1}} = \frac{2Ne^2}{3m_e}\left(\frac{E/N}{\Delta\varepsilon}\right)^2\left[\left(\frac{\varepsilon_k}{\bar{\varepsilon}_{k+1}}\right)^{1/2}\frac{\varepsilon_k}{\nu_k}\right] \tag{15b}$$

This is balanced by the energy lost through elastic collisions:

$$\frac{\partial E_{el}}{\partial t} = -\sum_k \left[(a_k - \bar{a_k}) - (b_k - \bar{b_k})\right] n_k \Delta\varepsilon \tag{16}$$

and inelastic processes:

$$\frac{\partial E_{inel}}{\partial t} = \sum_{ksj}\Big[R_{sjk+m(sj)}\, n_{k+m(sj)} N_s m(sj) + R^{i}_{sjk+m(si)}\, n_{k+m(si)} N_s m(si) - R^{l}_{sjk-m(sj)}\, n_{k-m(sj)} N^{j}_{s} m(sj)\Big]\Delta\varepsilon \tag{17}$$

The energy lost through all processes is subtracted from the energy gained, and the resulting number is the energy balance. A good energy balance will typically result from a problem with a good choice of binwidth and at least a four order of magnitude drop between highest and lowest number densities.

Transport Properties

Once a good distribution function has been calculated, quantities such as electron drift velocity, electron diffusion coefficient, and electron average energy, which are based on the electron EDF, can be calculated. Drift velocity is calculated by the following (15:380):

$$v_d = \frac{1}{En_e}\frac{\partial E_g}{\partial t} \tag{18}$$

where $\frac{\partial E_g}{\partial t}$ is calculated above in equation (15) for energy balance, E is the electric field and n_e the electron number density. The diffusion coefficient is calculated by (15:380):

$$D_f = \frac{1}{3N}\sqrt{\frac{2}{m_e}}\sum_k \frac{\varepsilon_k f_k \Delta\varepsilon}{\sum_s q_s \sigma_{sk}} \tag{19}$$

where q_s is the concentration of species s, σ_{sk} is the elastic collision cross-section for species s at bin k, and f_k is the reduced electron distribution function:

$$f_k = \frac{1}{\Delta\varepsilon\sqrt{\varepsilon}}\left(\frac{n_k}{n_e}\right) \tag{20}$$

Average energy is calculated as (15:380):

$$\langle\varepsilon\rangle = \sum_k \varepsilon_k n_k / n_e \tag{21}$$

Units used are cm s^{-1} for drift velocity, cm^2 s^{-1} for the diffusion coefficient, and eV for average energy.

Characteristic energy is another quantity which can be determined experimentally. It is a function of the diffusion coefficient and drift velocity, two other moments of the Boltzmann Equation, and provides a rough estimate of what the electron kinetic energy will be in the given distribution. It can be calculated by (15:380):

$$\varepsilon_{characteristic} = \frac{D_f}{\mu} = \frac{ED_f}{v_d} \tag{22}$$

where μ is the mobility of the electrons. This can be calculated by dividing the electron drift velocity by the applied electric field. Units of the characteristic energy are in electron volts.

Design Assumptions

Two major premises went into the design of the MEGABOLTZ code, which ended up dictating how the distribution function would be calculated. The first premise was using the implicit-explicit Euler method to transform the system of differential equations into a system of linear equations, instead of using a preprogrammed differential equation solver, like what can be found in the IMSL libraries. Preprogrammed differential equation programs cannot simultaneously address both the effects of superelastic and electron-electron collisions and provide the robustness required for this investigation (10). The implicit-explicit approach not only allows continuous updating of the time-dependent collision rates (such as electron-electron) during the calculation, but also minimizes the number of operations required each iteration.

The other design premise was in choosing LU-decomposition as the technique for solving the system of linear equations. This choice was made primarily due to the method's speed. Gaussian elimination takes n^3 operations, where n is the size of the matrix being manipulated (12:30-31), and an upper-Hessenberg reduction, as in NOMAD, will usually take around $\frac{5}{6}n^3$ operations for large n (12:368). A typical LU-decomp for a large matrix will typically cost around $\frac{1}{3}n^3$ operations (12:33). Also, both the IMSL version 9.2 and version 10.0 libraries

possess several routines for solving linear systems of equations by LU-decomposition, making it unnecessary to write these routines from scratch.

A brief explanation of LU-decomposition is in order, to better understand why this method was chosen. A typical system of linear equations can be expressed in matrix-vector format as follows:

$$\underline{A}\mathbf{x} = \mathbf{b} \tag{23}$$

where $\underline{A}$ is a square matrix of order n, $\mathbf{x}$ is a vector consisting of unknown variables, and $\mathbf{b}$ is the vector of known values. The matrix $\underline{A}$ can be represented by:

$$\underline{A} = \underline{L} \cdot \underline{U} \tag{24}$$

where L is a lower-triangular matrix (a matrix consisting of zeros above the diagonal) and $\underline{U}$ is an upper-triangular matrix (it has zeros below the diagonal). This turns our linear system into:

$$(\underline{L} \cdot \underline{U})\mathbf{x} = \mathbf{b} \tag{25}$$

We can now regroup this system of equations as follows:

$$\underline{L}\mathbf{y} = \mathbf{b} \tag{26a}$$

$$\underline{U}\mathbf{x} = \mathbf{y} \tag{26b}$$

We are given $\mathbf{b}$, so we can perform forward substitution with $\underline{L}$ to find $\mathbf{y}$. Once $\mathbf{y}$ is found, we can perform backward substitution with $\underline{U}$ to find $\mathbf{x}$.

Both version 9.2 and version 10 IMSL decompsition routines used Crout's method with partial pivoting to decompose the matrix. Crout's method assumes that both $\underline{L}$ and $\underline{U}$ can be stored in a single matrix, which will typically use the same storage space that $\underline{A}$ used to occupy (a process which destroys $\underline{A}$). The decomposed matrix $\underline{LU}$ is then calculated by columns from left to right, each column being calculated from top to bottom. The diagonal elements of $\underline{L}$ are then set equal to the diagonal elements of $\underline{U}$ during element calculation. Rowwise permutations of this matrix to get the largest elements on the diagonal insure the stability of this method during calculation (12:34-35).

V. Analysis and Discussion

Techniques Used For Code Validation

MEGABOLTZ was first run for elastic interactions only. For both constant collision frequency and constant cross-section, analytic solutions exist in the form of Maxwellian and Druyvesteyn distributions. An initial distribution similar to its intended final form was used to measure how quick the algorithm converged on the "correct" answer, and how far away it got from the "correct" answer in the process. Next, an arbitrary initial distribution was used for input, to see if, in each case, the algorithm would generate the appropriate distribution. A similar sequence of runs was performed for electron-electron interactions only.

After these validations were run, nitrogen gas was used, elastic and inelastic interactions only were assumed, and the EDF calculated. Data generated by the program (such as drift velocity and average energy) was then compared to values in the literature to measure algorithm accuracy. The time step was then varied to examine the convergence to a steady-state solution and the required number of time steps.

Constant Frequency Cases

First, all inelastic electron-neutral interactions and electron-electron interactions were turned off, and the MEGABOLTZ code specifically programmed to generate a distribution for an elastic collision frequency of 10^8 s^{-1}. For an E/N of 10^{-18} V-cm^2 (0.1 Townsends), gas pressure of 1 torr, gas temperature of 0 K, and electron number density of 10^{14} cm^{-3}, MEGABOLTZ successfully generated the Maxwellian EDF in Figure 4 over a 100-point grid of spacing 0.5 eV and a time step of 0.001 seconds, given a Maxwellian distribution for an initial guess. In Figure 5, a Druyvesteyn distribution using the same input parameters was used for an arbitrary initial guess. In this case, the distribution function still relaxed to Maxwellian form. The vertical scale of this figure is different from Figure 4 because of the inclusion of the initial distribution in the graph (the

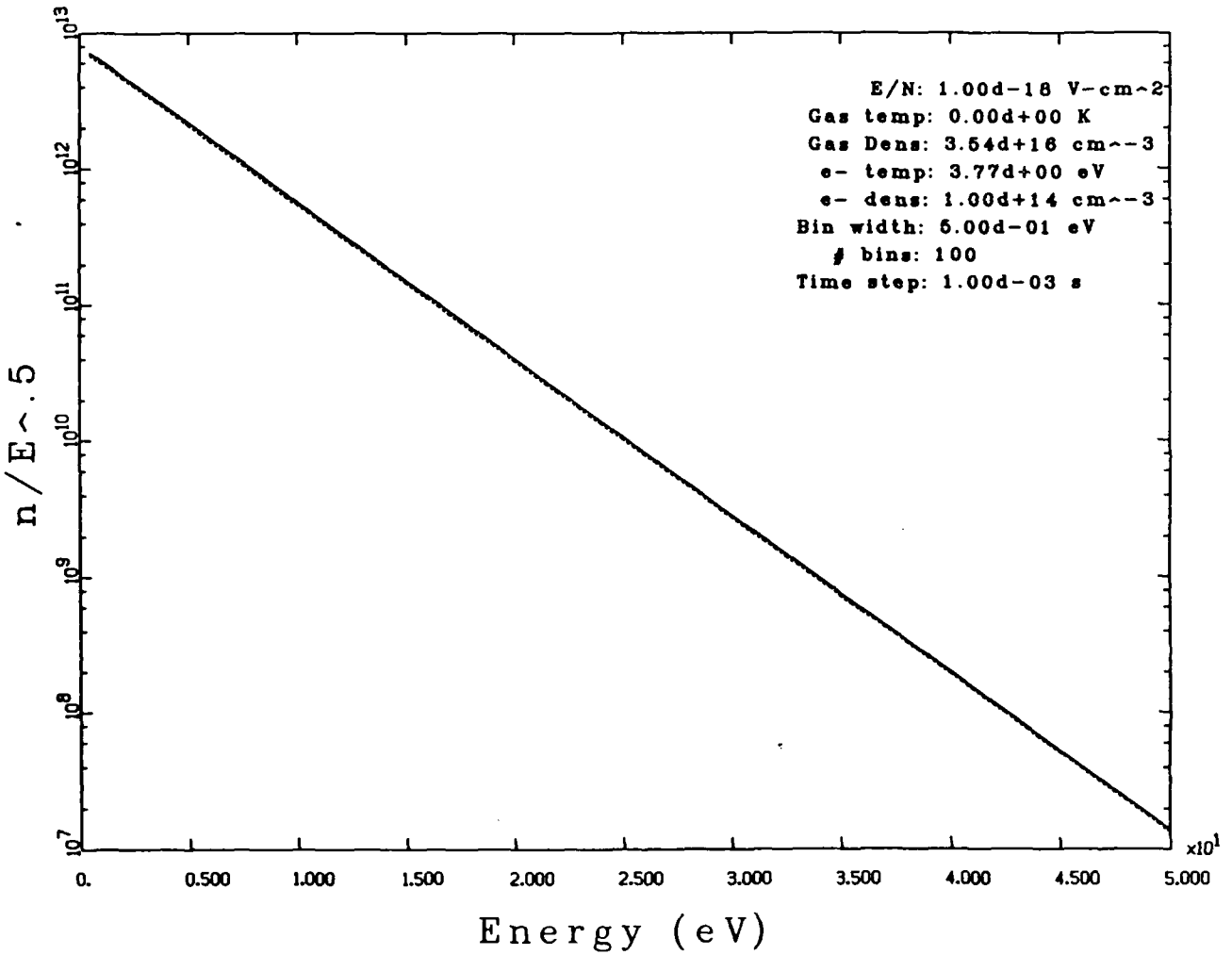


Figure 4: EDF for Constant Collision Frequency Using a Maxwellian Electron Distribution as an Initial Guess

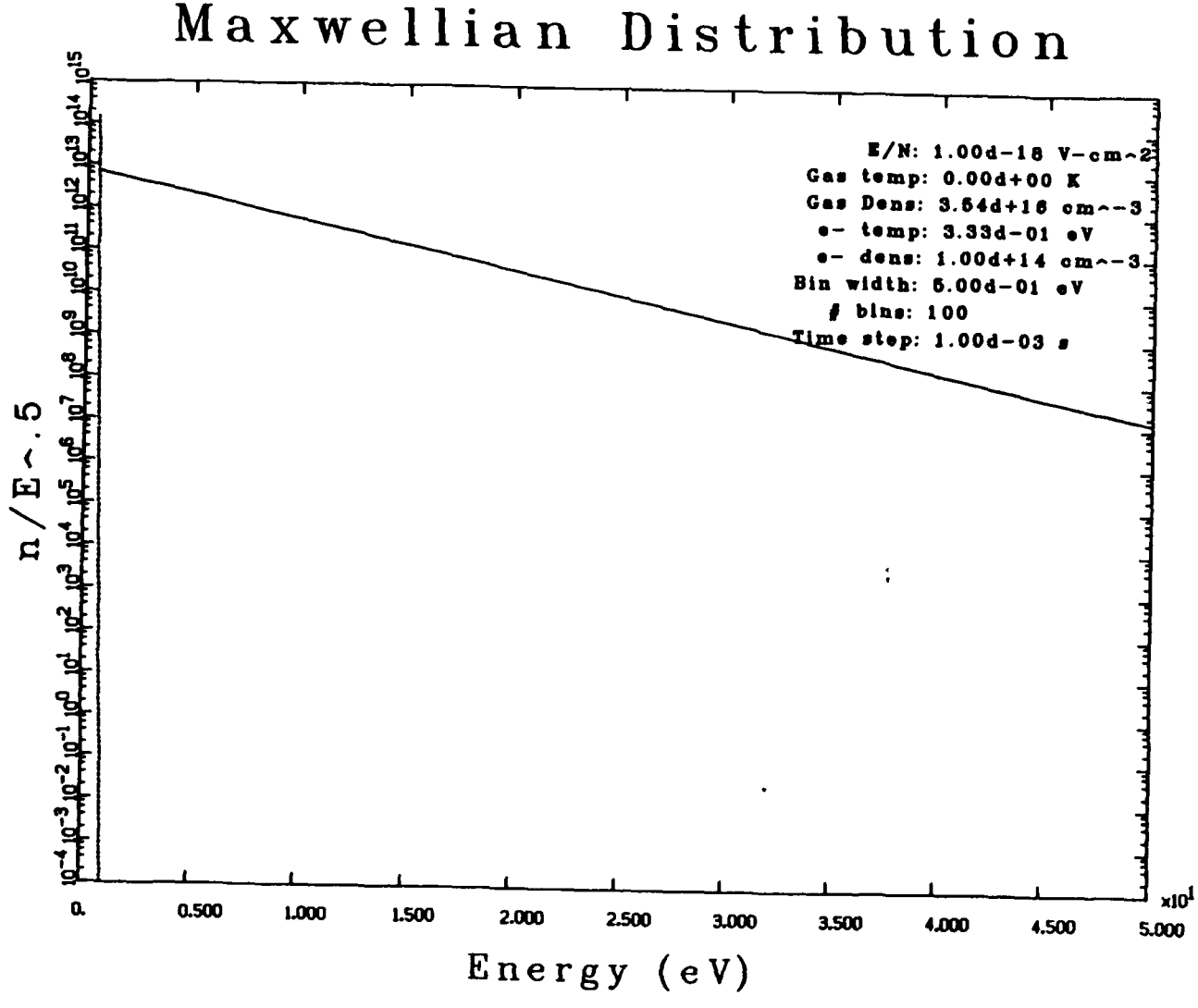


Figure 5: EDF for Constant Collision Frequency Using a Druyvesteyn Electron Distribution as an Initial Guess

nearly-vertical dotted line at the far left of the graph). In both runs, the electron temperature indicated in the parameters block in the upper right-hand corner is for the initial distribution.

Constant Cross-Section Cases

Next, MEGABOLTZ was specifically programmed to generate an EDF using an elastic collision cross-section of 8×10^{-16} cm^2 for all energies. In figure 6, MEGABOLTZ successfully reproduced a Druyvesteyn distribution when it was used as an initial guess, for the input parameters listed in the upper right-hand corner of the figure. Using a Maxwellian distribution for the same input parameters as an arbitrary initial guess, the distribution function still relaxed to Druyvesteyn form, as shown in Figure 7.

Electron-Electron Interactions

MEGABOLTZ's handling of electron-electron interactions was verified by running the program with all electron-neutral interactions turned off. Since this renders the algorithm in equation (10) fully, explicit, the time step needed to be limited to a fraction of the electron-electron collision frequency to insure computational stability. This point was verified by running MEGABOLTZ with several different time steps during this phase of validation. Time steps greater than 10^{-11} seconds were found to cause the calculation to go unstable, driving the generated distribution negative at certain energies. Therefore, a time step of 10^{-11} seconds was used for this phase.

The final EDF calculated by the program in this case should be a Maxwellian. With this in mind, MEGABOLTZ was first run with a Druyvesteyn distribution of electrons for an initial distribution function. The initial distribution was generated assuming an E/N of 1 Townsend, and an elastic electron-neutral collision cross-section of 8×10^{-16} cm^2. This had the effect of giving the program an arbitrary electron distribution, and tested it to see if the electrons relaxed into the functional form which by theory they should. When this test case was run, maximum number of

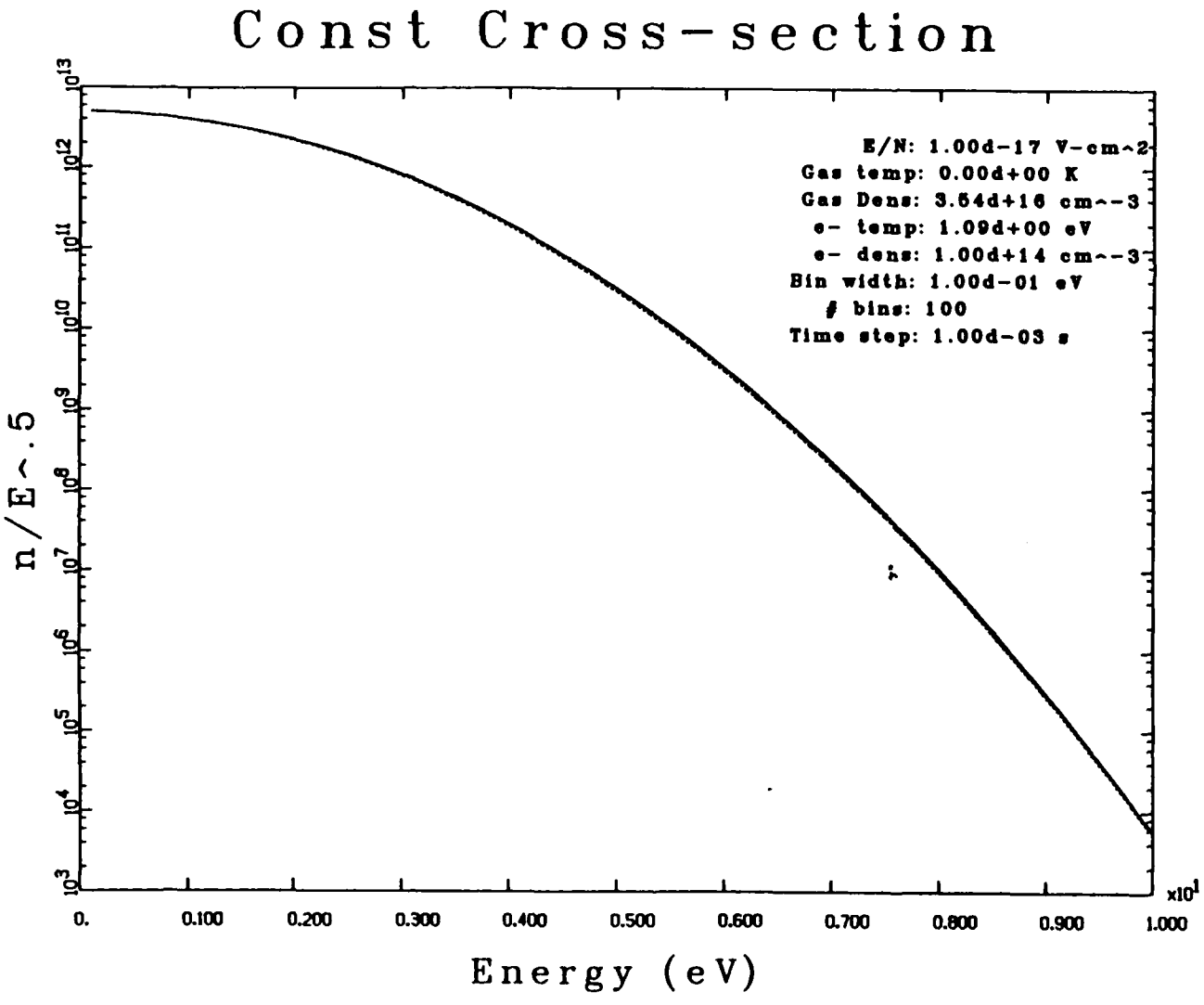


Figure 6: EDF for Constant Cross-section Using a Druyvesteyn Electron Distribution as an Initial Guess

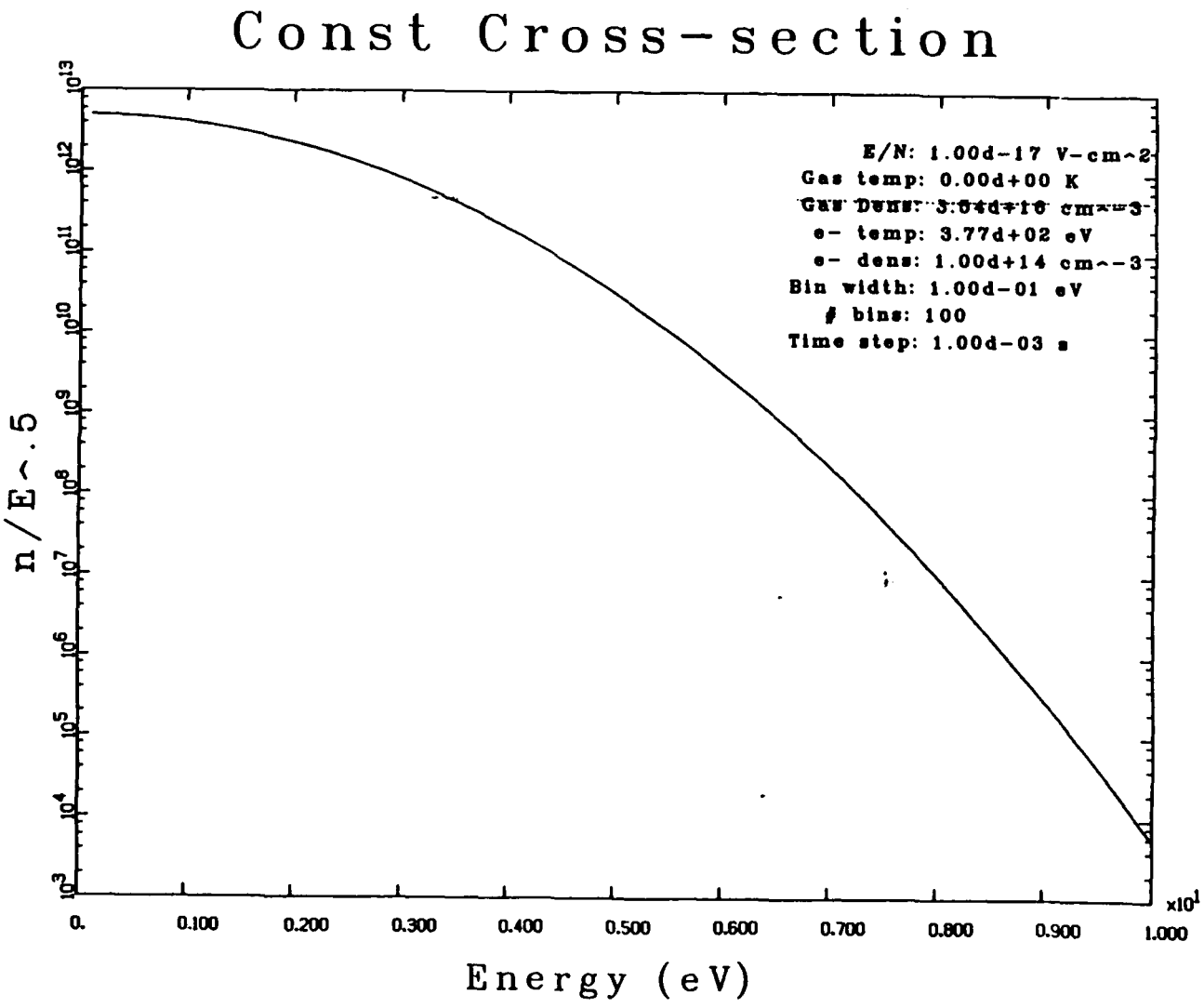


Figure 7: EDF for Constant Cross-section Using a Maxwellian Electron Distribution as an Initial Guess

iterations (5000) was reached before convergence tolerance (10^{-4}) was met. However, as shown in Figure 8, the EDF calculated by MEGABOLTZ was indeed Maxwellian in shape, as it should be under electron-electron dominant interactions. Figure 8 clearly shows the relaxation of the distribution function from its initial form (the dotted line) to final form (the straight solid line). When a Maxwellian distribution (generated assuming E/N = 0.1 Townsends and collision frequency = 10^8 s^{-1})was used as the initial guess, MEGABOLTZ converged to the answer shown in Figure 9 in one iteration. The final EDF calculated is virtually indistinguishable from the initial guess, again as theory predicts.

Nitrogen

At this point, an attempt was made to reproduce electron-transport data reported by Rockwood and Greene in 1980 for N_2 gas. Using the NOMAD code, with superelastic, ionization, and electron-electron interactions turned off, they used one atmosphere of N_2 at 300 K, and calculated electron drift velocity and average energy with a distribution function over 250 bins of width 0.01 eV for E/N values ranging from 0.5 - 4.0 x 10^{-16} V-cm^2 (5 - 40 Townsends) (15:383). MEGABOLTZ was run using these conditions to generate moments, which were then compared against both similar quantities calculated by NOMAD and experimental drift velocity and diffusion data for $N_2$to verify MEGABOLTZ's accuracy.

First, I compared drift velocity and average energy of the EDF respectively with the values reported by Rockwood and Greene (15:383). Tables 2 and 3 show the results of these calculations, which were performed using a 100-bin energy axis of width 0.05 eV. Graphs were also generated at this time, showing drift velocity (Figure 10) and average energy (Figure 11) as a function of E/N. The values calculated by MEGABOLTZ are generally in good agreement with those calculated by NOMAD. What disparities exist between them can be attributed to MEGABOLTZ performing its calculation on a coarser grid (100 bins of width 0.05 eV, as opposed

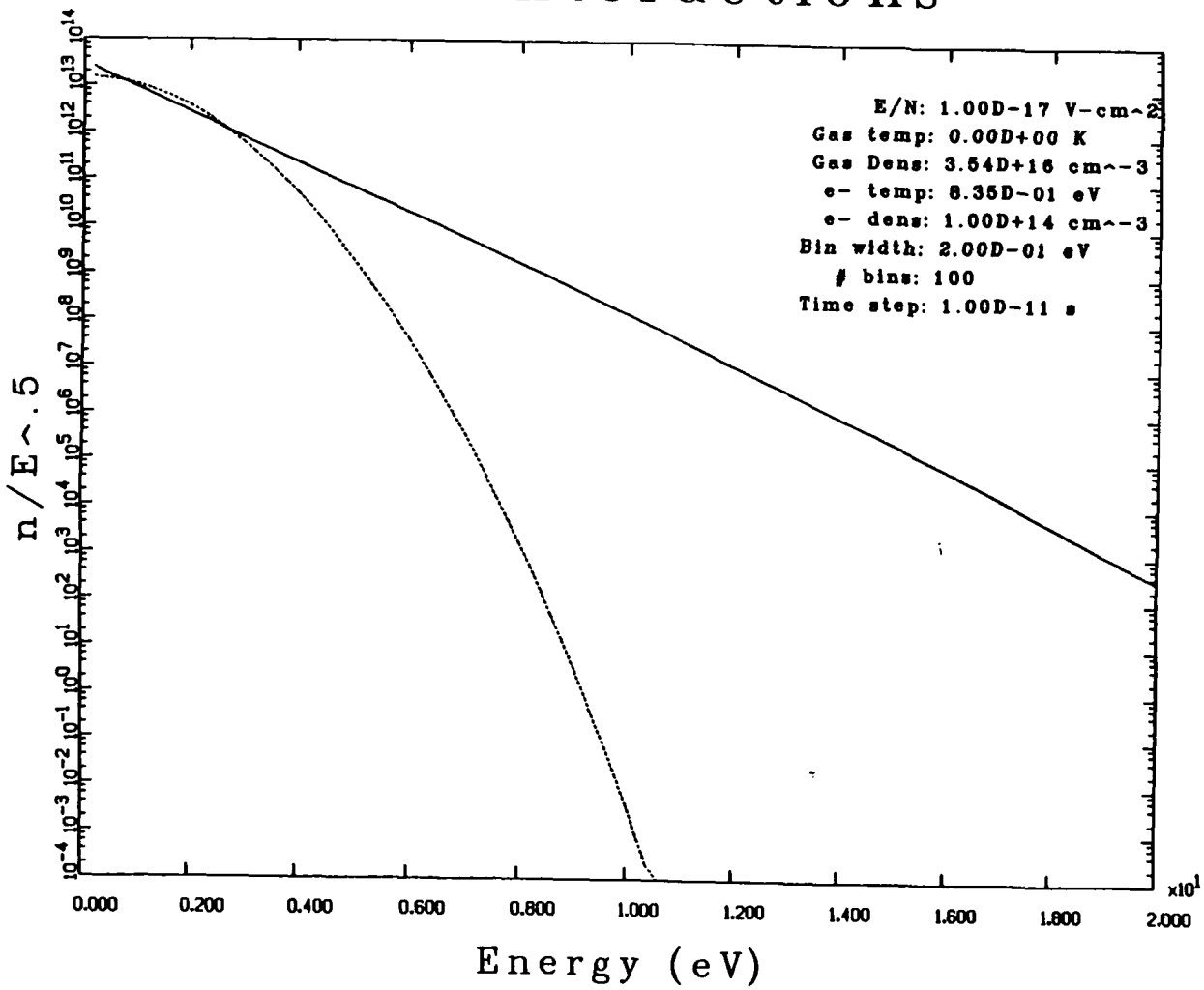


Figure 8: EDF for Electron-electron Interactions Using a Druyvesteyn Electron Distribution as an Initial Guess

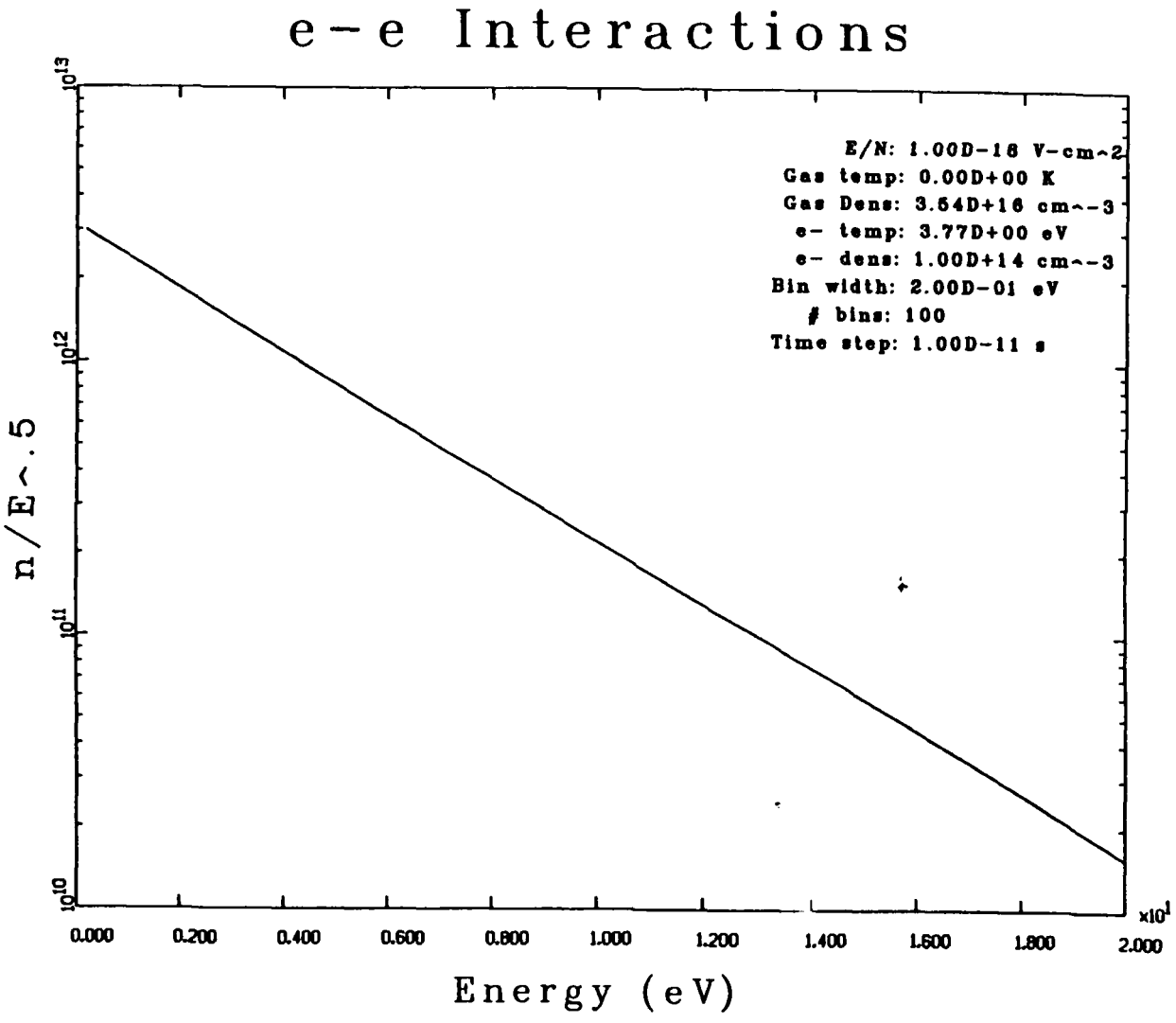


Figure 9: EDF for Electron-electron Interactions Using a Maxwellian Electron Distribution as an Initial Guess

Table 2:
Electron Drift Velocity in N_2 gas as a Function of E/N

E/N (Td)	v_d ($x10^6$ cm s^{-1}) Calculated	v_d ($x10^6$ cm s^{-1}) Rockwood/Greene	Pct Error
5	1.0736	1.09	1.51
10	1.7726	1.78	0.42
15	2.4383	2.43	0.34
20	3.0712	3.06	0.37
30	4.2303	4.21	0.48
40	5.2263	5.26	0.64

Gas Temp (K): 300
Gas Pressure (Torr): 760
Electron Number Density (cm^{-3}): 10^{14}
Number of Bins: 100
Bin Width (eV): 0.05
Time Step (s): 0.001

Table 3:
Electron Average Energy in N_2 gas as a Function of E/N

E/N (Td)	<ε> (eV) Calculated	<ε> (eV) Rockwood/Greene	Pct Error
5	0.8804	0.846	4.07
10	0.9869	0.959	2.91
15	1.0278	1.006	2.17
20	1.0544	1.033	2.07
30	1.1006	1.079	2.00
40	1.1613	1.134	2.41

Gas Temp (K): 300
Gas Pressure (Torr): 760
Electron Number Density (cm^{-3}): 10^{14}
Number of Bins: 100
Bin Width (eV): 0.05
Time Step (s): 0.001

to NOMAD's grid of 250 bins of width 0.01 eV). This problem appears to be most pronounced at low E/N, where the EDF is changing rapidly with respect to energy.

The above calculations were repeated on MEABOLTZ for the NOMAD grid and the results are presented in Tables 4 and 5. Going to a finer grid reduces the previously-noted disparities for low E/N. Remaining differences can be attributed to minor deviations between the cross-section sets used. It is interesting to note the agreement in drift velocities until an E/N of 30

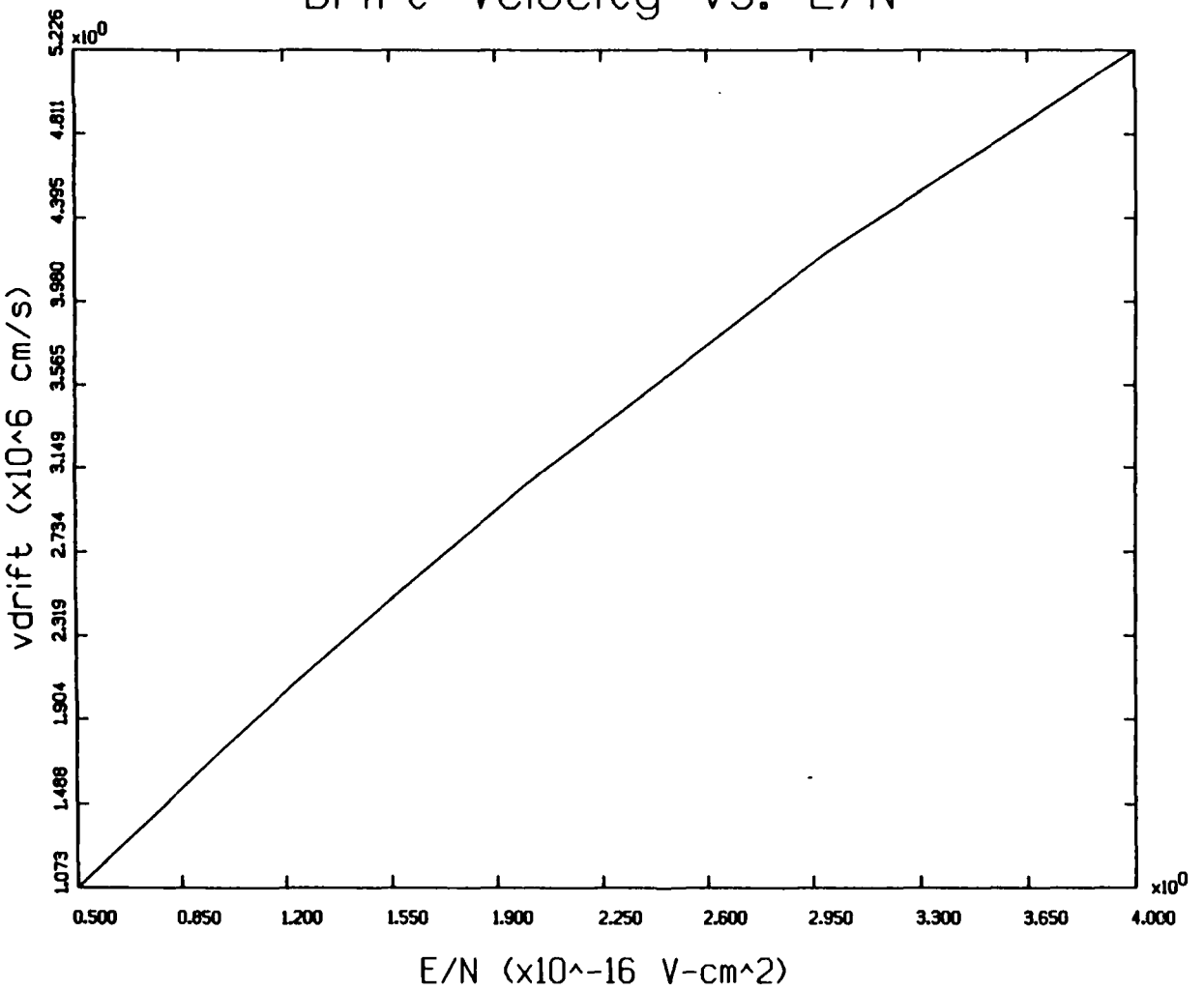


Figure 10: Computed Electron Drift Velocities for Nitrogen as a Function of E/N

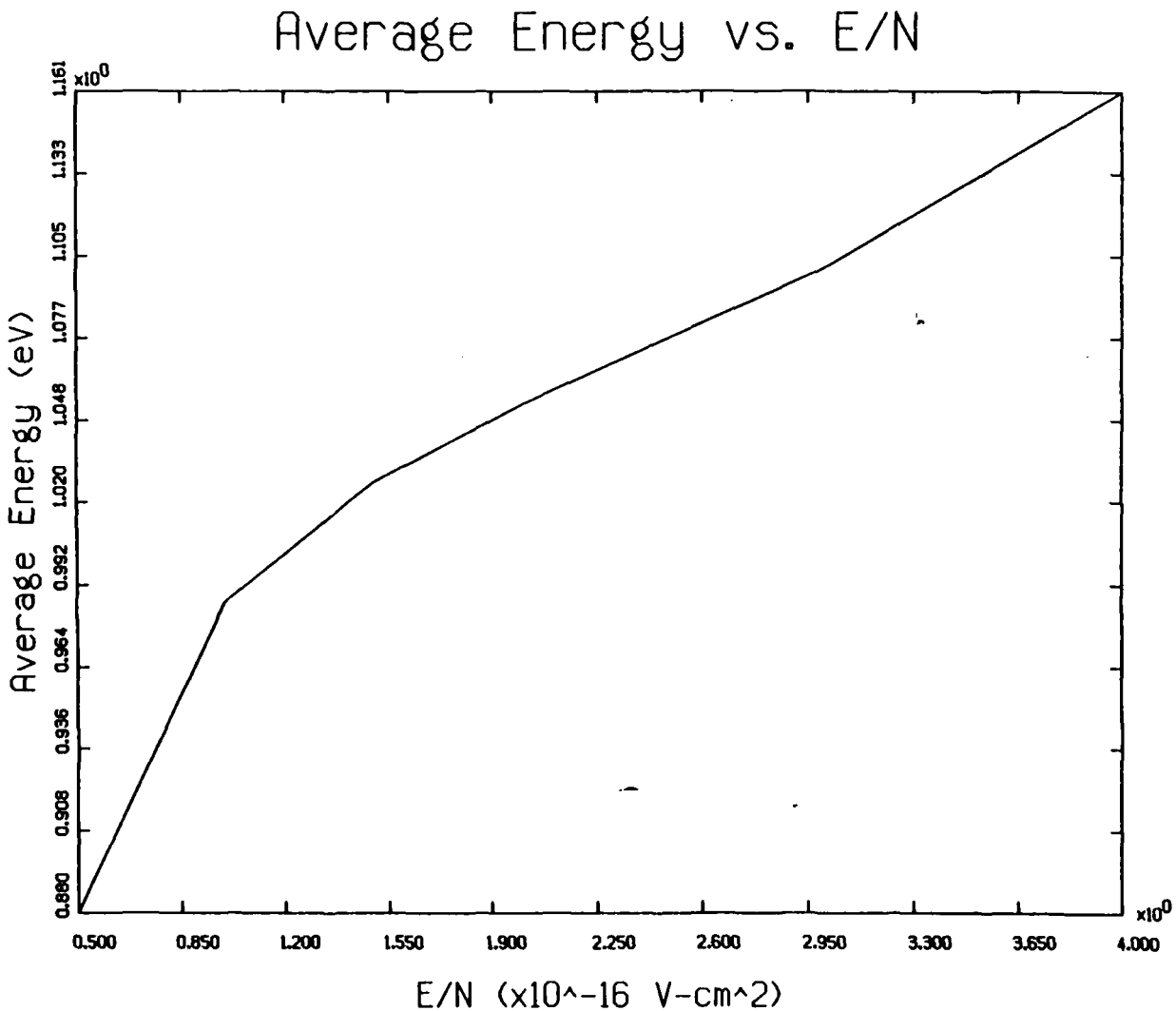


Figure 11: Computed Electron Average Energy for Nitrogen as a Function of E/N

Table 4:
Electron Drift Velocity in N_2 gas as a Function of E/N

E/N (Td)	v_d ($x10^6$ cm s^{-1}) Calculated	v_d ($x10^6$ cm s^{-1}) Rockwood/Greene	Pct Error
5	1.0793	1.09	0.98
10	1.7712	1.78	0.49
15	2.4266	2.43	0.14
20	3.0474	3.06	0.41
30	4.1549	4.21	1.31
40	5.0425	5.26	4.14

Gas Temp (K): 300
Gas Pressure (Torr): 760
Electron Number Density (cm^{-3}): 10^{14}
Number of Bins: 250
Bin Width (eV): 0.01
Time Step (s): 0.001

Table 5:
Electron Average Energy in N_2 gas as a Function of E/N

E/N (Td)	<ε> (eV) Calculated	<ε> (eV) Rockwood/Greene	Pct Error
5	0.8572	0.846	1.32
10	0.9672	0.959	0.85
15	1.0101	1.006	0.41
20	1.0378	1.033	0.47
30	1.0820	1.079	0.28
40	1.1225	1.134	1.01

Gas Temp (K): 300
Gas Pressure (Torr): 760
Electron Number Density (cm^{-3}): 10^{14}
Number of Bins: 250
Bin Width (eV): 0.01
Time Step (s): 0.001

Townsends. At this point, there was insufficient room on the energy axis for the distribution function to properly tail off. This could best be seen in the energy balancing for these last two cases, which was only 1 in 100. The calculated average energy from MEGABOLTZ, however, agreed very well with the figures calculated by NOMAD for all values of E/N.

To correct for the above problem, the binwidth was doubled and the same calculations were repeated. The opprotunity was also taken to generate a plot of the EDF, which is presented in

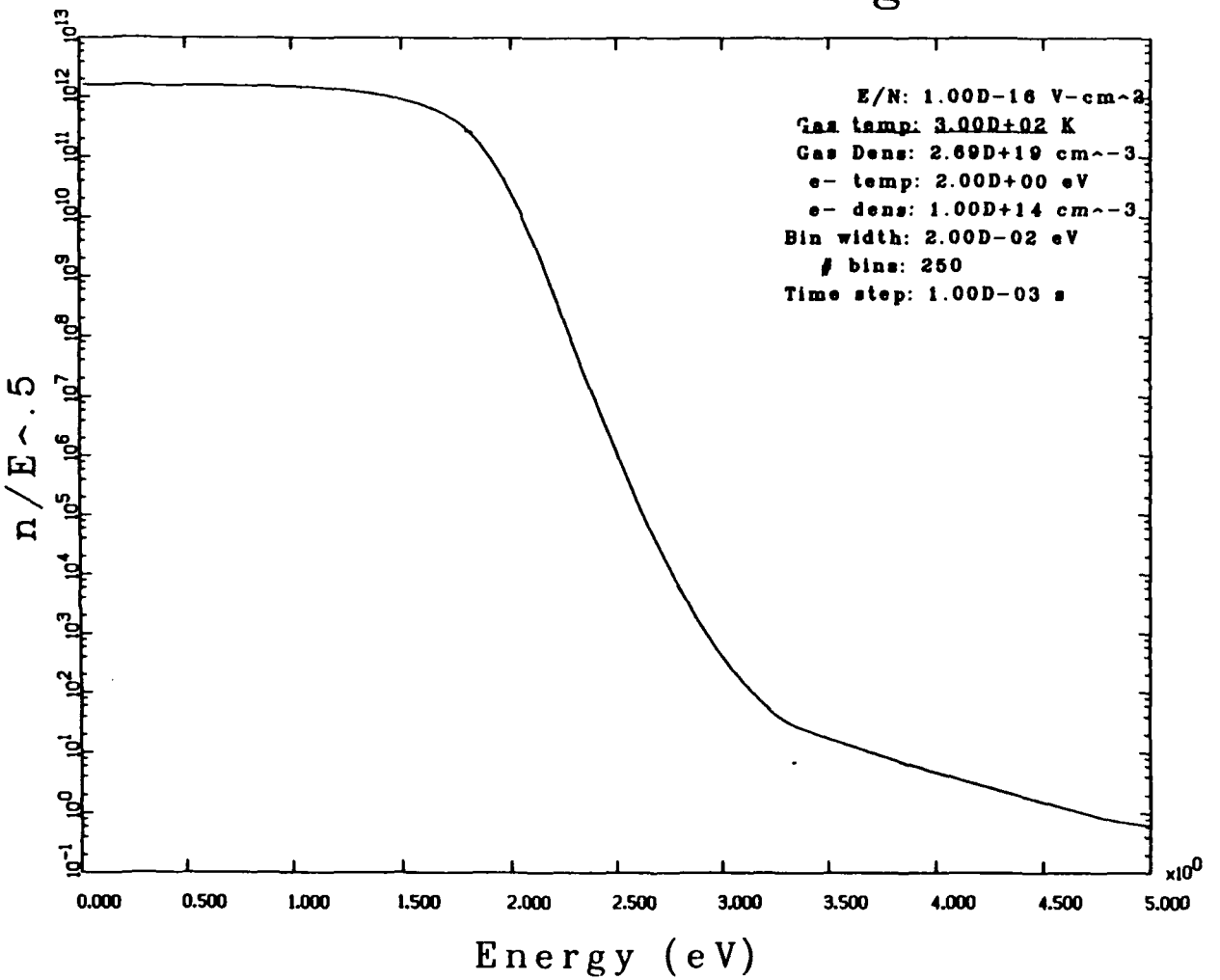


Figure 12: Computed Electron EDF for Nitrogen

Table 6:
Experimental vs. Computed Electron Drift Velocity in N_2 Gas

E/N (Td)	v_d (calc) (10^6 cm s^{-1})	v_d (expt) (10^6 cm s^{-1})	Pct Error	v_d (R/G) (10^6 cm s^{-1})	Pct Error
5	1.0848	1.0950	0.93	1.0900	0.48
10	1.7800	1.8380	3.16	1.7800	0.00
15	2.4389	2.4767	1.52	2.4300	0.36
20	3.0653	3.0900	0.80	3.0600	0.17
30	4.2146	4.1700	1.07	4.2100	0.11
40	5.2085	5.1800	0.55	5.2600	0.99

Gas Temp (K): 300
Gas Pressure (Torr): 760
Electron Number Density (cm^{-3}): 10^{14}
Number of Bins: 250
Bin Width (eV): 0.02
Time Step (s): 0.001

Table 7:
Experimental vs. Computed Characteristic Energies in N_2 Gas

E/N (Td)	D_f (calc) (cm^2 s^{-1})	D_f/μ (calc) (eV)	D_f/μ (expt) (eV)	Pct Error	D_f/μ (R/G) (eV)	Pct Error
5	640.46	0.7932	0.744	6.61	0.786	0.92
10	655.16	0.9890	0.932	6.12	0.984	0.51
15	655.64	1.0835			1.084	0.05
20	653.53	1.1458			1.146	0.02
30	647.62	1.2387			1.237	0.13
40	643.19	1.3273			1.312	1.16

Gas Temp (K): 300
Gas Pressure (Torr): 760
Electron Number Density (cm^{-3}): 10^{14}
Number of Bins: 250
Bin Width (eV): 0.02
Time Step (s): 0.001

Figure 12. As expected, increasing the maximum energy corrected the problem with the drift velocities at 30 and 40 Townsends. Drift velocity and characteristic energy data were compared to data from Huxley and Crompton (7:628) on these runs. The results are shown in Tables 6 and 7. Characteristic energy data in Huxley and Crompton only went to 10 Townsends, which meant only the MEGABOLTZ runs at 5 and 10 Townsends could be compared against actual experiments.

As can be seen in Table 7, there is a six-plus percent error between the calculated and experimental characteristic energies. It is appropriate to note that NOMAD could do no better when determining characteristic energy, as MEGABOLTZ values were invariably within one percent or so of NOMAD values. Drift velocities, however, generally show excellent agreement with experimental data. These particular runs were repeated on a coarser energy grid, halving the number of bins while doubling the binwidth. This should result in slightly greater errors between calculated and experimental data, since the binwidth has been doubled for the same maximum energy. Results are presented in Tables 8 and 9. Characteristic energy errors between MEGABOLTZ and experiment are slightly better, as well as drift velocities.

Next, errors caused by premature truncation of the distribution function were determined. The number of bins was held constant at 250, and the bin widths varied from 0.005 eV to 0.08 eV. The effects of this variation on percent error in number of iterations, drift velocity, average energy, and the value of the reduced distribution f(e) in the last bin are shown in Table 10, and Figures 13, 14, and 15. There appeared to be an optimum binwidth of 0.02 eV for the particular input parameters listed in Table 6, which gave answers in close agreement to the NOMAD calculations reported by Rockwood and Greene (15:383). The drastic difference between the runs of binwidth 0.005 eV and 0.01 eV is explained by the fact that for the first value listed, the distribution function had insufficient space to properly tail off, and thus did not approach zero at max energy. The errors for the extremely wide binwidths can be explained by the fact the bin width is too wide to resolve structure in the calculated EDF

Errors caused by zoning the energy axis were tested next. The number of bins and binwidth were varied so as to run the energy axis out to 2.5 eV. Results of this test for number of iterations, drift velocity, average energy, and value of the reduced distribution in the last bin are tabulated in Table 11. In the case of a fixed max energy, there also appeared to be an optimum choice of binwidth and number of bins (125 bins at 0.02 eV/bin) which gave best agreement with

Table 8:
Experimental vs. Computed Electron Drift Velocity in N_2 Gas

E/N (Td)	v_d (calc) (10^6 cm s^{-1})	v_d (expt) (10^6 cm s^{-1})	Pct Error	v_d (R/G) (10^6 cm s^{-1})	Pct Error
5	1.0933	1.0950	0.16	1.0900	0.30
10	1.7966	1.8380	2.25	1.7800	0.92
15	2.4624	2.4767	0.58	2.4300	1.32
20	3.0946	3.0900	0.15	3.0600	1.12
30	4.2520	4.1700	1.97	4.2100	0.99
40	5.2497	5.1800	1.35	5.2600	0.20

Gas Temp (K): 300
Gas Pressure (Torr): 760
Electron Number Density (cm^{-3}): 10^{14}
Number of Bins: 125
Bin Width (eV): 0.04
Time Step (s): 0.001

Table 9:
Experimental vs. Computed Characteristic Energies in N_2 Gas

E/N (Td)	D_f (calc) (cm^2 s^{-1})	D_f/μ (calc) (eV)	D_f/μ (expt) (eV)	Pct Error	D_f/μ (R/G) (eV)	Pct Error
5	641.81	0.7887	0.744	6.01	0.786	0.34
10	656.29	0.9815	0.932	5.32	0.984	0.25
15	656.63	1.0748			1.084	0.85
20	654.42	1.1364			1.146	0.83
30	648.34	1.2291			1.237	0.64
40	643.84	1.3182			1.312	0.47

Gas Temp (K): 300
Gas Pressure (Torr): 760
Electron Number Density (cm^{-3}): 10^{14}
Number of Bins: 125
Bin Width (eV): 0.04
Time Step (s): 0.001

values reported by Rockwood and Greene, who used a 250-point, 0.01 eV grid for their calculations (15:390).

Finally, for the same input parameters as in Tables 4 and 5, using 250 bins of width 0.01 eV, MEGABOLTZ's speed of convergence was tested for timesteps ranging from 10^2 to 10^{-11} seconds. Number of iterations required for each timestep is presented in Figure 16. This

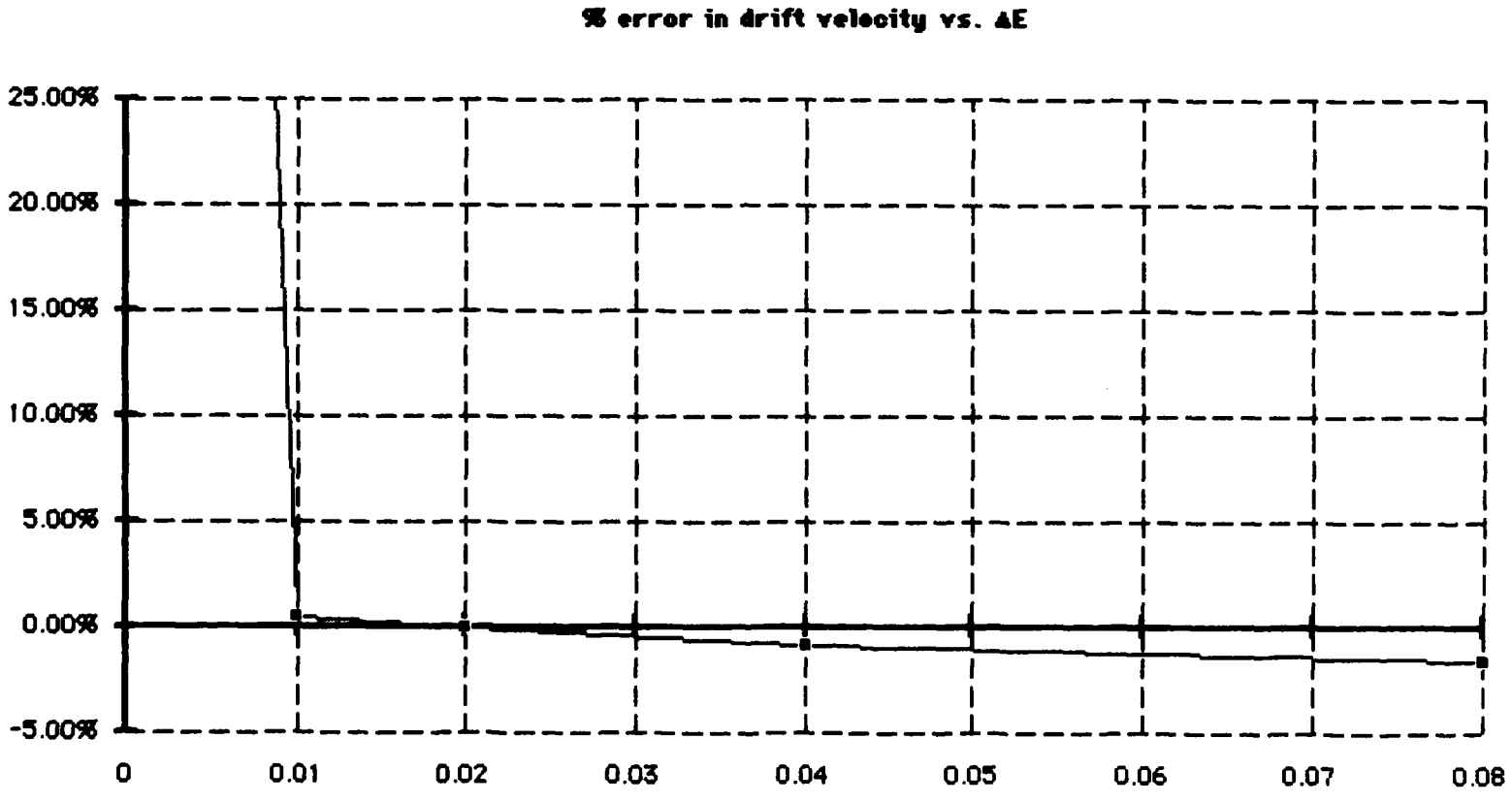


Figure 13: Computed Electron Drift Velocities as a Function of Bin Width

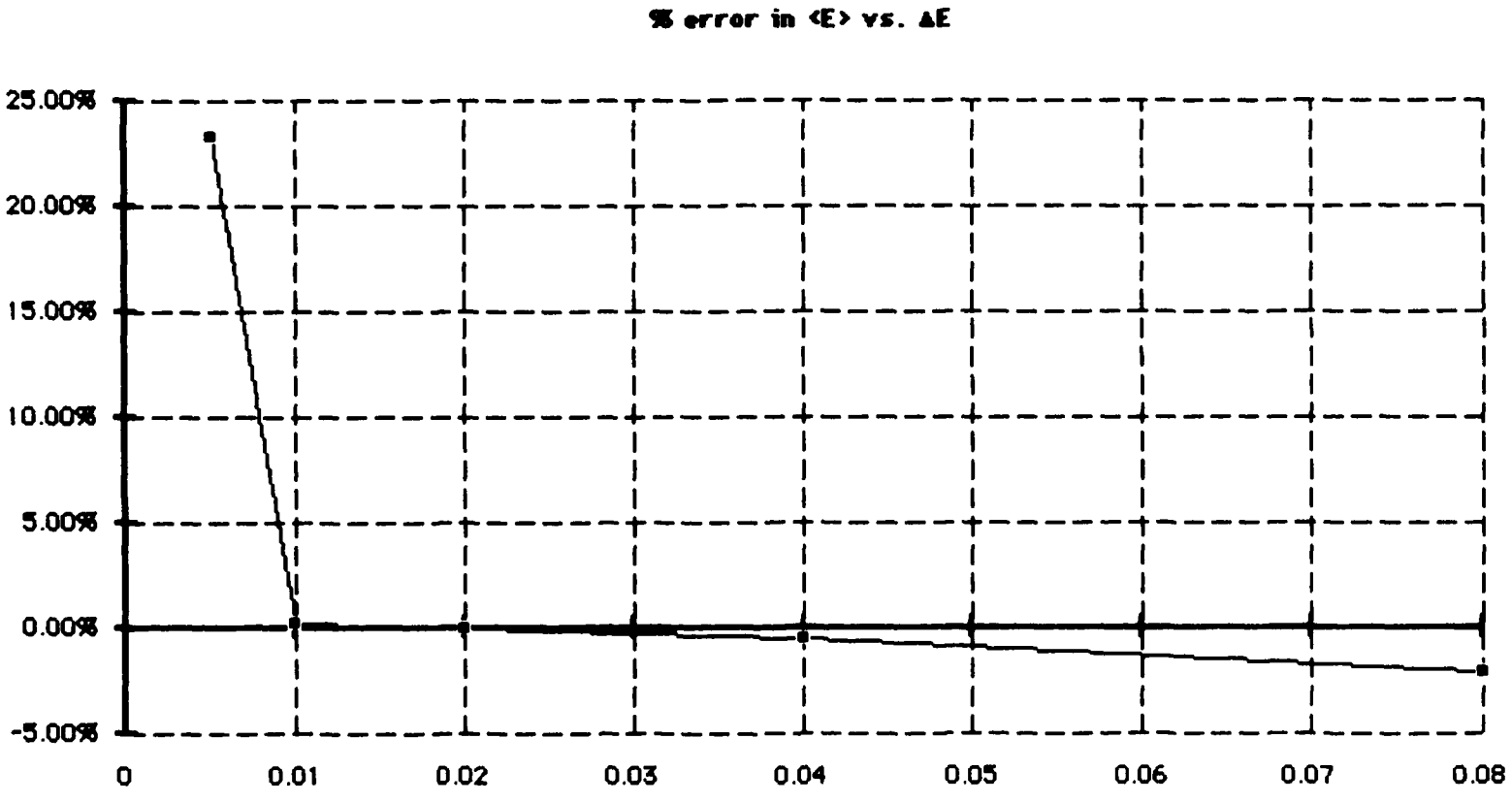


Figure 14: Computed Electron Average Energy as a Function of Bin Width

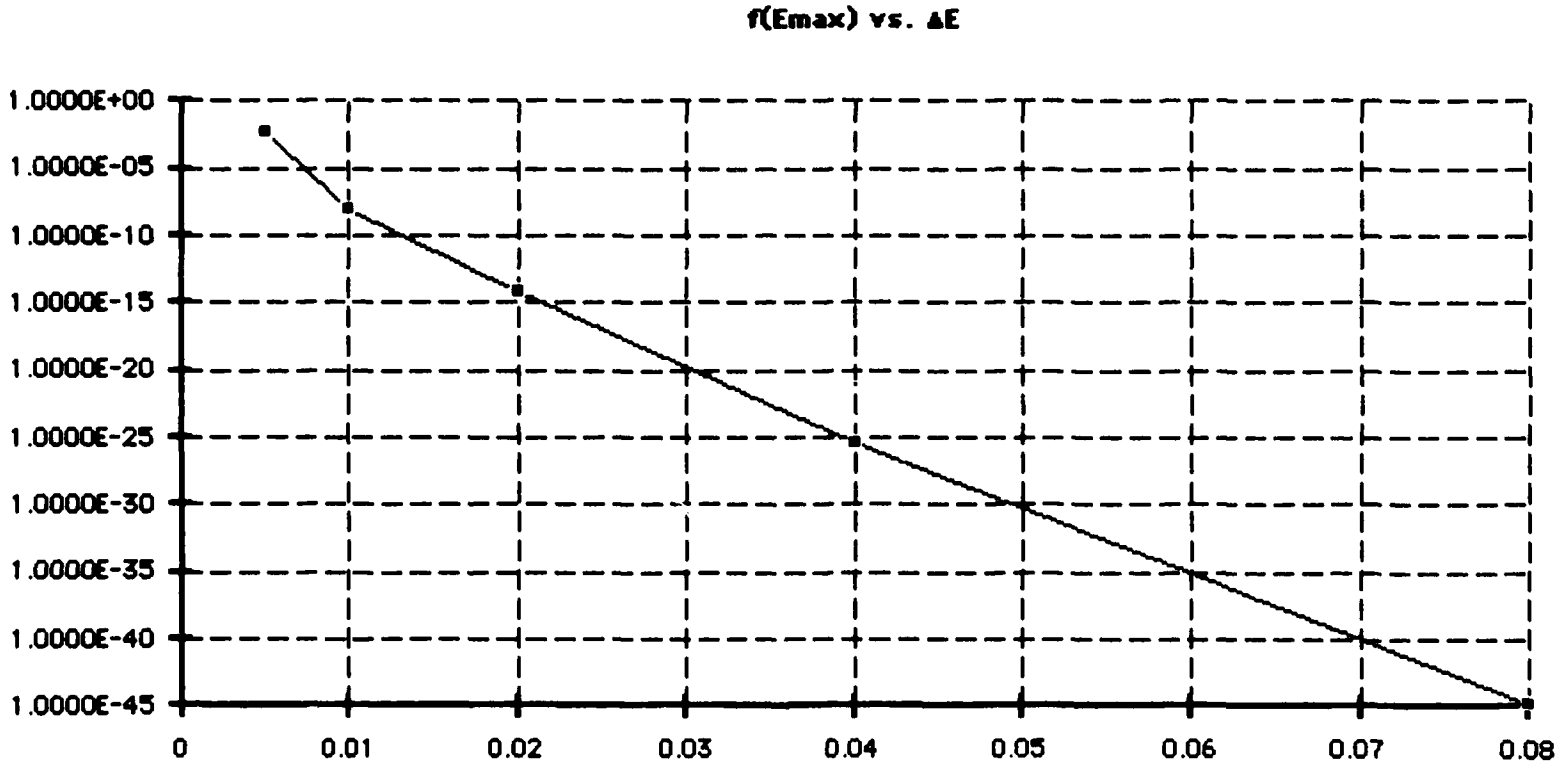


Figure 15: f(Emax) as a Function of Bin Width

Table 10:
Truncation Error for a 250-point Energy Axis as a Function of Bin Width

$\Delta\varepsilon$ (eV)	Iterations	v_d ($\times 10^6$ cm s^{-1})	$\langle\varepsilon\rangle$ (eV)	$f(\varepsilon_{max})$
0.005	2	0.1199	0.7448	$5.1682*10^{-3}$
0.01	3	1.7712	0.9672	$1.0031*10^{-8}$
0.02	4	1.78	0.9698	$6.3594*10^{-15}$
0.04	4	1.7966	0.9747	$4.3535*10^{-26}$
0.08	7	1.8083	0.9907	$2.3357*10^{-45}$

Gas Temp (K): 300
Gas Pressure (Torr): 760
Electron Number Density (cm^{-3}): 1

E/N ($V\text{-}cm^2$): 10^{-16}
Number of Bins: 250
Time Step (s): 0.001

Table 11:
Calculation Errors Due To Energy Axis Zoning

$\Delta\varepsilon$ (eV)	Bins	Iterations	v_d ($\times 10^6$ cm s^{-1})	$\langle\varepsilon\rangle$ (eV)	$f(\varepsilon_{max})$
0.01	250	3	1.7712	0.9672	$1.0031*10^{-8}$
0.0125	200	3	1.7795	0.9662	$1.2256*10^{-8}$
0.02	125	3	1.78	0.9698	$1.9694*10^{-8}$
0.025	100	3	1.7789	0.9726	$2.5031*10^{-8}$

Gas Temp (K): 300
Gas Pressure (Torr): 760
Electron Number Density (cm^{-3}): 1

E/N ($V\text{-}cm^2$): 10^{-16}
Time Step (s): 0.001

---

sequence of runs also establishes the utility of the fully-implicit method by showing convergence for a wide range of time steps. Drift velocity and average energy were examined, and no variance in their value was noted for all timesteps investigated except 10^{-11} seconds. This is likely due to one or both of the following problems: first, the Boltzmann Equation gives answers which do not make sense physically for cases where the timestep is much less than the inverse of the collision

frequency of the dominant interaction; and second, for extremely small time steps, roundoff errors during calculation become non-trivial.

Speed of Algorithm

Speed of calculation was one of the original design criteria for MEGABOLTZ. This section will explain how fast MEGABOLTZ ran on the BLACKBIRD and GALAXY computers at AFIT and why.

The BLACKBIRD mainframe was where I performed the bulk of code development for MEGABOLTZ. BLACKBIRD consisted of a VAX 11/785, running BSD UNIX version 4.3 in a multiuser environment. There were some special programming considerations when running on this mainframe. For starters, the two routines for matrix decomposition and linear-equation solving were called from the version 10.0 IMSL FORTRAN libraries. Also, the code was not optimized at compile time, allowing use of the symbolic debugger DBX. The biggest disadvantages of running on BLACKBIRD was lack of speed and heavy system loading due to other jobs. When doing numerically-intensive runs in these situations, such as pure electron-electron interactions, or matrices larger than 100x100 elements, MEGABOLTZ would take several minutes to several hours of real time to execute.

The GALAXY mainframe was where I eventually performed the most numerically-intensive calculations. It consisted of a cluster of 10 ELXSI CPUs. One 'side' of this cluster ran an EMBOS environment, while the other side of GALAXY ran BSD UNIX version 4.3. I worked on the UNIX side, due to prior experience with that operating system. When running properly, GALAXY would typically run several times faster than BLACKBIRD for a similar number of users and processes. Unfortunately, the UNIX side of the GALAXY possessed only a version 9.2 double-precision IMSL library. This created a problem when the code was ported over from BLACKBIRD, because (as mentioned previously) the original version of the code invoked two IMSL version 10.0 routines to perform LU-decomposition and back-substitution. Fortunately, all that was necessary was to

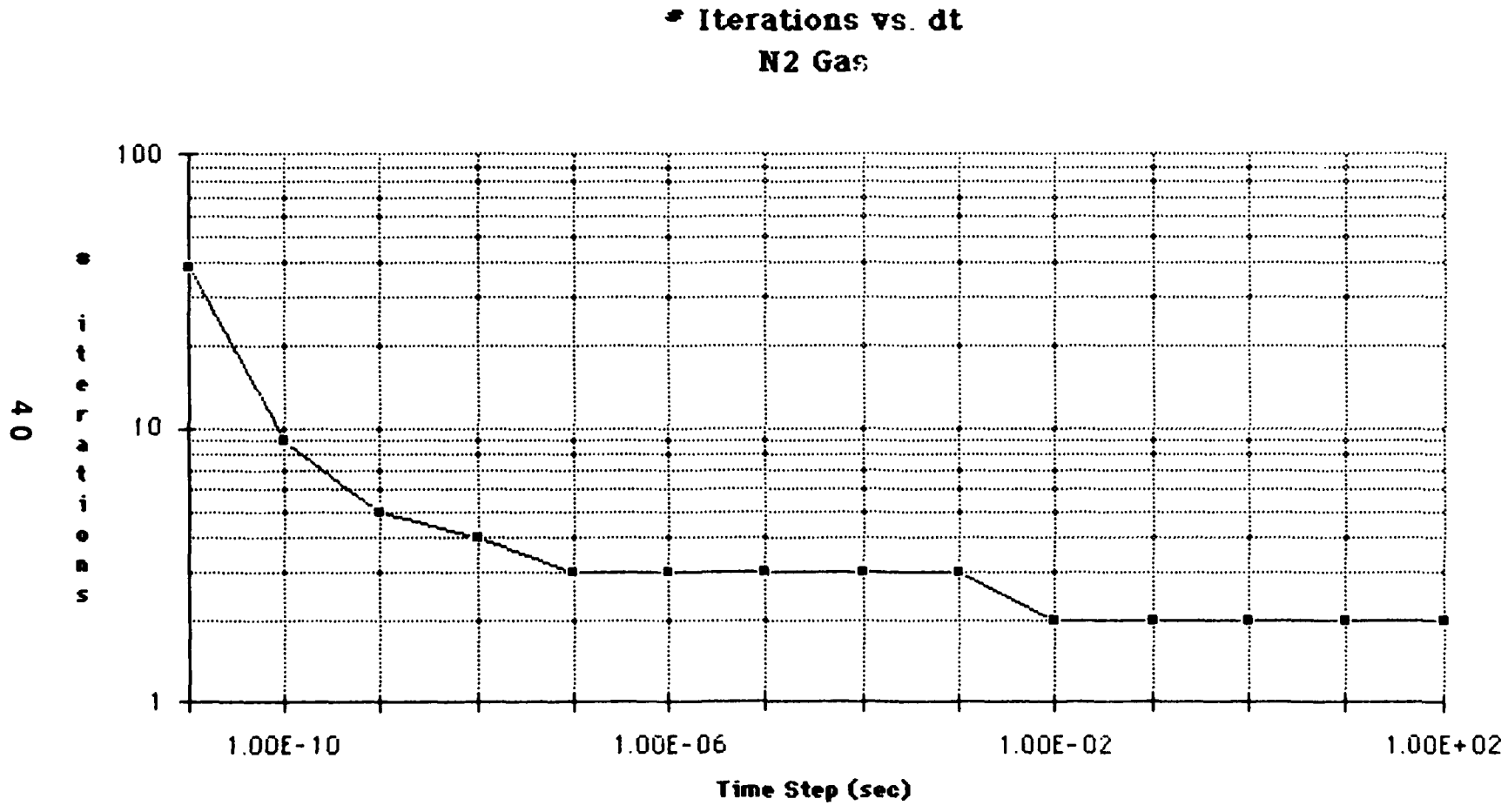


Figure 16: Number of Iterations vs. Timestep

rewrite the two subroutine calls in question and declare some extra variables that the older subroutine calls expected.

MEGABOLTZ was run twice on GALAXY during time trials. The first run, presented in Table 12, was performed for N_2 gas on a 250-bin energy axis of bin width 0.02 eV. Input parameters used were the same as for the 10-Townsend validation run for N_2, shown in Figure 10 above. The second run, featured in Table 13, was an exact repeat of the program run which generated Figure 8, a run which considered electron-electron interactions only. Both runs took 4.9 seconds of CPU time to run, but with vastly different clock times. This was due to the fact a much smaller grid was used for the electron-electron problem. Even so, it was suprising to note that the time saved in this problem by not decomposing a matrix was almost exactly balanced by the fact the program went the full designed number of time iterations.

CPU timing runs are unfortunately unavailable for BLACKBIRD, as its implementation of the CPU/clock timing function was buggy and caused MEGABOLTZ to crash the first time it was called.

Table 12:
Run Times for GALAXY Version of MEGABOLTZ for N_2 gas

Program Reference	Clock Time (sec)	CPU Time (sec)
Thru LOOKUP:	4.4000	2.0000
Thru COEFF:	10.2000	0.2000
Thru EECOLL:	0.0000	0.0000
Thru Decomposition:	439.1000	0.1000
Thru Time iteration:	22.8000	0.0000
Thru BALANCE:	24.2000	0.1000
Thru SAVEDATA:	5.0000	0.5000
Thru GRAFIT:	26.1000	1.9000
Total:	531.9000	4.9000

Gas Temp (K): 300

Gas Pressure (Torr): 760

Electron Number Density (cm^{-3}): 1

E/N ($V\text{-}cm^2$): 10^{-16}

Number of Bins & Width: 250, at 0.02eV/bin

Time Step (s): 0.001

Table 13:
Run Times for GALAXY Version of MEGABOLTZ for Electron-electron Interactions

Program Reference	Clock Time(sec)	CPU Time (sec)
Thru LOOKUP:	3.5000	1.9000
Thru COEFF:	0.2000	0.1000
Thru EECOLL:	6.1000	0.1000
Thru Time iteration:	3895.9000	0.4000
Thru BALANCE:	8.7000	0.0000
Thru SAVEDATA:	2.4000	0.5000
Thru GRAFIT:	25.6000	1.8000
Total:	3942.3999	4.9000

E/N ($V\text{-}cm^2$): 10^{-16}

Electron Number Density (cm^{-3}): 1

Number of Bins & Width: 100, at 0.2eV/bin

Time Step (s): 10^{-11}

E/N value listed was used for calculation of the initial guess distribution.

VI. Conclusions, Suggestions and Recommendations

Conclusions

In conclusion, a computer program for solving the time-dependent Boltzmann Equation was developed and tested. The program used a different algorithm for calculating electron behavior which eliminated low-energy instabilities, and a more efficient computational technique for faster run times. The program gave results which agreed well with previous Boltzmann equation programs and experimental data.

While the primary design objectives were met during the course of this thesis, I feel there is still room for improvement in certain aspects of both the equations and program. I will now talk about four of these areas, which in my opinion merit further study.

Fully Implicit Electron-Electron Interactions

For cases where electron-electron interactions dominate, an implicit-explicit algorithm is presently used. This limits the time step which can be used for calculating an EDF, by arguments presented in the last section. Another method can be considered for handling this case. Consider equation (10), only this time invert the $[1 + \underline{I}h]$ matrix . Next, multiply the inverse with the $[1-\underline{C}h]$ matrix , then LU-decompose the product. This would allow the treatment of electron-electron interactions using a fully-implicit algorithm. Although computationally less efficient due to the matrix inversion, multiplication, and decomposition taking place each iteration, this treatment would allow a much greater range of time steps to be used. The present algorithm, however, appears to be sufficient enough to handle these cases.

Variable Energy Binwidth

MEGABOLTZ, as presently designed, assumes the mesh spacing (or alternatively, bin width) is constant with respect to energy. If the slope of the calculated distribution as a function of

energy has little to no structure, this is not a liability. If, however, the computed distribution has areas of rapid variation in energy interspersed with areas where the function is hardly varying at all, there can be problems. To accurately resolve such structure, a narrow binwidth would have to be used, increasing the number of bins in energy space -- a move which would essentially be wasted over the parts of the function which are nearly flat, and would increase the size of the distribution function and coefficient matrices, causing the program to run slower. The solution to this problem is to implement some scheme in which the binwidth is either a function of energy or a function of the slope of the calculated distribution.

Bretagne has investigated varying binwidth as a function of energy. In his code, grid points on the energy axis are allocated logarithmically, on the assumption that the rapidly-changing portions of the distribution function are at low energies (3:814). In practice, this is a very good approximation since the assumption it is based upon is true for most all distribution functions of interest. Another technique of interest is presented by Nickel as part of a laser modeling code (11:14-19). A generic mesh-allocation scheme is used, where the density of mesh points is a function of the slope of the distribution function. This scheme will work with any distribution function, anywhere on the energy axis. Some computational overhead, for the equations which allocate the points on the energy axis, would be involved in implementing such a scheme.

In the present implementation of MEGABOLTZ, great care must be taken in the implementation of any variable-binwidth scheme. MEGABOLTZ relies on certain quantities which are dependent on binwidth, such as array offsets for inelastic and ionization processes (calculated by taking integer portion of [energy loss/$\Delta\mathcal{E}$]). At the least, it would be advisable to keep an array for $\Delta\mathcal{E}$, since it is now going to be a function of position along the energy axis.

Attachment Processes

In a plasma, there is the chance that electrons could be lost from the distribution function due to attachment processes. This interaction involves an electron hitting a neutral molecule,

which retains the electron and becomes a negative ion. The electron is not only lost from the EDF, but its energy is lost as well. This process is time-independent, and can in principle be treated similarly to other electron-neutral procedures such as inelastic collisions. The fact that electrons and energy is removed from the system also need to be accounted for during energy balancing.

User-defined source-loss terms

In a real- life discharge tube or ion source, there might be electron losses from collisions with the wall(s) of the vessel, depending on the exact problem being modeled. Also, depending on the physics behind the particular problem, there might be a source of electrons, such as an electron beam or a filament. A more robust version of MEGABOLTZ should support some form of source and/or loss terms somewhere in the calculation. For treatment of source functions, a good starting point would be Elliott and Greene (6:2948), followed by Bretagne (3:812). Electron losses could conceivably be modeled by assuming ambipolar diffusion through the walls of the discharge tube (2:2210, 3:813).

APPENDIX A: Full Derivation of Flux-Divergent Boltzmann Equation for All Electron-Neutral Collision Terms

The basic time-dependent Boltzmann equation for electrons is:

$$\frac{\partial f}{\partial t} + (\mathbf{v}\cdot\nabla_r)f + (\mathbf{a}\cdot\nabla_v)f = \left(\frac{\delta f}{\delta t}\right)_{collisions} \qquad (A-1)$$

If we assume the distribution function f(r,v,t) is constant in space, and the velocity distribution of the electrons is spherical (to a first approximation), Equation 1 simplifies to: (8:368)

$$\frac{\partial f}{\partial t} + \frac{eE}{m_e}\frac{\partial f}{\partial v} = \left(\frac{\delta f}{\delta t}\right)_{collisions} \qquad (A-2)$$

To account for pertubations in the electron velocity distribution, we now use a two-term expansion in spherical harmonics (8:368):

$$f(v,\theta) = f_0(v) + f_1(v)\cos\theta \qquad (A-3)$$

Equation 2 now becomes:

$$\left[\frac{\partial}{\partial t} + \frac{eE}{m_e}\frac{\partial}{\partial v}\right]\left[f_0(v) + f_1(v)\cos\theta\right] = \left(\frac{\delta f}{\delta t}\right)_{collisions} \qquad (A-4)$$

If we assume DC electric fields, this equation can be rewritten into a form identical to Equations 53 and 54' in Holstein (8:378-379):

$$\frac{\partial f}{\partial t} = -\frac{4(eE/m_e)^2}{3u^{1/2}}\frac{\partial}{\partial u}\left(\frac{u}{NQ}\frac{\partial f_0}{\partial u}\right) - \frac{2m_e}{M}\frac{\partial}{\partial u}\left(u^2 NQ f_0\right) \qquad (A-5)$$

where $u = v^2$. If we let $\mathcal{E} = m_e u/2$ and substitute as appropriate into the above equation, it can be rearranged into:

$$\frac{\partial f}{\partial t} = -\frac{4(eE/m_e)^2}{3(2\mathcal{E}/m_e)^{1/2}} m_e \frac{\partial}{\partial \mathcal{E}}\left(\frac{\mathcal{E}}{NQ}\frac{\partial f_0}{\partial \mathcal{E}}\right) - \frac{2m_e}{M}\frac{\partial}{\partial \mathcal{E}}\left(\mathcal{E}^2 NQ f_0\right) \qquad (A-6)$$

If we define J_{el} and J_f as fluxes of electrons in energy space due to momentum transfer collisions and external fields:

$$J_{el} = -\frac{2m_e}{M} \nu \varepsilon^{3/2} f_0 \qquad \text{(A-7a)}$$

$$J_f = \frac{2Ne^2(E/N)^2}{3(2m_e)^{1/2}} \frac{1}{\varepsilon^{1/2}} \left(\frac{\varepsilon}{Q} \frac{\partial f_0}{\partial \varepsilon} \right)$$

$$= \frac{\varphi}{\varepsilon^{1/2}} \left(\frac{\varepsilon}{Q} \frac{\partial f_0}{\partial \varepsilon} \right) \qquad \text{(A-7b)}$$

Equation A-6 simplifies to:

$$\frac{\partial f}{\partial t} = -\frac{\partial J_f}{\partial \varepsilon} - \frac{\partial J_{el}}{\partial \varepsilon} + \left(\frac{\delta f}{\delta t} \right)_{other} \qquad \text{(A-8)}$$

When this equation is finite-differenced, we want the electron fluxes to be nodal quantities. While energy, collision frequency, and collision cross-section are all nodal quantities, the reduced distribution f and the energy distribution n are centered quantities. Some algebraic manipulation is therefore necessary to rectify this problem. Let us first consider the finite-differencing of J_f, the flux due to the external field:

$$J_f(k) = \frac{\varphi}{(\varepsilon_k^-)^{1/2}} \left(\frac{f_{k+1} - f_k}{\Delta \varepsilon} \right) \qquad \text{(A-9)}$$

where

$$\varepsilon_k^- = \varepsilon_k - \frac{\Delta \varepsilon}{2}$$

The finite-differenced derivative of the above equation is thus:

$$\frac{\partial J_f}{\partial \varepsilon} = \frac{\varphi}{(\varepsilon_k^-)^{1/2}}\left[\left(\frac{\varepsilon_k}{\Delta\varepsilon Q_k}\right)f_{k+1} - \left(\frac{\varepsilon_k}{\Delta\varepsilon Q_k} + \frac{\varepsilon_{k-1}}{\Delta\varepsilon Q_{k-1}}\right)f_k + \left(\frac{\varepsilon_{k-1}}{\Delta\varepsilon Q_{k-1}}\right)f_{k-1}\right]$$

(A-10)

To get equation A-10 into terms of the energy distribution function n_k, we now multiply through by $(\varepsilon_k^-)^{1/2}$ to get:

$$\frac{\partial J_f(k)}{\partial \varepsilon} = \frac{\varphi}{\Delta\varepsilon}\left[\left(\frac{\varepsilon_k}{Q_k}\right)\frac{n_{k+1}}{(\varepsilon_{k+1}^-)^{1/2}} - \left(\frac{\varepsilon_k}{Q_k} + \frac{\varepsilon_{k-1}}{Q_{k-1}}\right)\frac{n_k}{(\varepsilon_k^-)^{1/2}} + \left(\frac{\varepsilon_{k-1}}{Q_{k-1}}\right)\frac{n_{k-1}}{(\varepsilon_{k-1}^-)^{1/2}}\right]$$

(A-11)

Consider Q_k. Since the collision frequency $\nu_k = NQ_k(2\varepsilon/m_e)^{1/2}$, we can solve this for Q_k and substitute into equation A-11, getting it in terms of energy, collision frequency, and number density:

$$\frac{\partial J_f(k)}{\partial \varepsilon} = \frac{\varphi N}{\Delta\varepsilon^2}\left(\frac{2}{m_e}\right)^{1/2}\left[\left(\frac{\varepsilon_k^{3/2}}{\nu_k}\right)\frac{n_{k+1}}{(\varepsilon_{k+1}^-)^{1/2}} - \left(\frac{\varepsilon_k^{3/2}}{\nu_k} + \frac{\varepsilon_{k-1}^{3/2}}{\nu_{k-1}}\right)\frac{n_k}{(\varepsilon_k^-)^{1/2}} + \left(\frac{\varepsilon_{k-1}^{3/2}}{\nu_{k-1}}\right)\frac{n_{k-1}}{(\varepsilon_{k-1}^-)^{1/2}}\right]$$

(A-12)

After performing some algebraic simplification, we then recover the promotion and demotion rates a_k and b_{k+1} for external fields:

$$a_k = \frac{2Ne^2}{3m_e}\left(\frac{E/N}{\Delta\varepsilon}\right)^2\left[\left(\frac{\varepsilon_k}{\varepsilon_k^-}\right)^{1/2}\frac{\varepsilon_k}{\nu_k}\right] \qquad \text{(A-13a)}$$

$$b_{k+1} = \frac{2Ne^2}{3m_e}\left(\frac{E/N}{\Delta\varepsilon}\right)^2\left[\left(\frac{\varepsilon_k}{\varepsilon_{k+1}^{-}}\right)^{1/2}\frac{\varepsilon_k}{\nu_k}\right] \qquad \text{(A-13b)}$$

Development of the finite-differenced elastic collision flux J_{el} proceeds exactly as in Rockwood, Appendix A (14:2356-2358).

A problem with Rockwood's original finite-differencing scheme was that an instability would occur in the first few bins. This was most pronounced when an analytic distribution was used as the initial guess, and the program using Rockwood's differencing scheme attempted to reproduce the initial guess as quickly as possible. The number density of the first bin would typically be over twenty percent lower than in the initial guess, when ideally it should not have moved at all. This error would typically damp out within the first five bins. The differencing scheme detailed in this appendix for external field terms greatly reduces this problem, and provides excellent reproducibility of an analytic distribution used as an initial guess.

APPENDIX B: Derivation of Electron-Electron Interaction Terms

A complete derivation is given in Appendix B of Reference 13. This appendix will be a (hopefully) brief summary of the important points of this derivation.

The time rate of change of electron number density for electron-electron interactions only can be written as follows:

$$\frac{\partial n}{\partial t} = \alpha n_o^2 F(\mathcal{E},t) \qquad \text{(B-1)}$$

where

$$\alpha = \frac{2\pi}{3} e^4 \left(\frac{2}{m_e}\right)^{1/2} \ln \Lambda \qquad \Lambda = \frac{(kT/4\pi n_o e^2)^{1/2}}{2e^2/m_e v^2}$$

$$F(\mathcal{E},t) = 2\mathcal{E}^2 \frac{\partial^2 f}{\partial \mathcal{E}^2}\left(\frac{1}{n_o}\frac{\partial \varphi}{\partial \mathcal{E}}\right) + \frac{1}{n_o}\frac{\partial f}{\partial \mathcal{E}}\left(\mathcal{E}\frac{\partial \varphi}{\partial \mathcal{E}} + \varphi\right) + 3\mathcal{E}^{1/2} f^2$$

$$\varphi(\mathcal{E},t) = 3\int_0^{\mathcal{E}} n\,dx - \frac{1}{\mathcal{E}}\int_0^{\mathcal{E}} x n\,dx$$

$$f(\mathcal{E}) = \mathcal{E}^{-1/2}\, n/n_o \qquad n_o = \int_0^{\infty} n(\mathcal{E})\,d\mathcal{E}$$

This turns out to be the Coulomb collision integral, represented as a momentum-space divergence of electron current density. In flux-divergent form, this can be represented as:

$$\frac{\partial n}{\partial t} = -\frac{\partial J_{ee}}{\partial \mathcal{E}} \qquad \text{(B-2)}$$

where

$$J_{ee} = \alpha\left[P\left(\frac{n}{2\varepsilon} - \frac{\partial n}{\partial \varepsilon}\right) - Q\,n\right]_{\varepsilon}^{\varepsilon}$$

$$P(\varepsilon,t) = 2\varepsilon^{1/2}\int_0^{\varepsilon} x\,n(x,t)dx + 2\varepsilon\int_0^{\varepsilon} x^{-1/2}n(x,t)dx$$

$$Q(\varepsilon,t) = 3\varepsilon^{-1/2}\int_0^{\varepsilon} n(x,t)dx$$

By finite-differencing equation 2 along the energy axis as before, and rearranging terms, we get:

$$A_{kj} = \alpha\left\{\left(\varepsilon_{k+1}^{-1/2}H_{k+1,j} + \varepsilon_k^{-1/2}H_{k,j}\right)\left(\varepsilon_j u_k^{+} - .75\right) + \left[(1-H_{k,j})\varepsilon_{k+1} + (1-H_{k-1,j})\varepsilon_k\right] u_k^{+}\varepsilon_k^{-1/2}\right\} \qquad \text{(B-3a)}$$

$$B_{k+1,j} = \alpha\left\{\left(\varepsilon_{k+1}^{-1/2}H_{k+1,j} + \varepsilon_k^{-1/2}H_{k,j}\right)\left(\varepsilon_j u_k^{-} + .75\right) + \left[(1-H_{k,j})\varepsilon_{k+1} + (1-H_{k-1,j})\varepsilon_k\right] u_k^{-}\varepsilon_k^{-1/2}\right\} \qquad \text{(B-3b)}$$

where

$$u_k^{\pm} = \frac{1}{\Delta\varepsilon} \pm \frac{.25}{\varepsilon_k + \Delta\varepsilon/2} \qquad H_{k,j} = \begin{cases} 0, & k<j \\ 1, & k\geq j \end{cases}$$

A_{kj} represents the rate, in cm^3/sec, that electrons lose energy from bin k to bin k-1 while exciting electrons from bin j to bin j+1. B_{kj} represents a similar quantity, except that electrons in bin k go to bin k+1 while de-exciting electrons from bin j to bin j-1. Both these matrices must observe three properties: they must conserve particles, they must conserve energy, and they

must reproduce a Maxwellian distribution in a steady-state situation. Let us first look at energy conservation. The time rate of change of the electron energy density is:

$$\frac{\partial E}{\partial t} = \sum_{k=1}^{K} \varepsilon_k \frac{\partial n}{\partial t} \Delta\varepsilon = \sum_{kl} (A_{kj} - B_{kj}) n_k n_j (\Delta\varepsilon)^2 \qquad \text{(B-4)}$$

If energy is to be conserved, this equation must equal zero. This implies:

$$\sum_{kl} (A_{kj} - B_{jk}) = 0 \qquad \text{(B-5)}$$

For this to be true, $A_{kj} - B_{jk}$ must be an antisymmetric matrix. Therefore, energy conservation can be assured by setting:

$$A_{kj} = B_{jk} \qquad \text{(B-6)}$$

From a physical interpretation of what should occur at the lowest and highest energies, particle conservation can be guaranteed by setting:

$$0 = A_{j1} = B_{jK} = A_{Kj} = B_{1j} \qquad \text{(B-7)}$$

No electrons are lost off the energy axis, nor do they appear from beyond it.

To insure $\frac{\partial n_k}{\partial t} = 0$ when n_k is a Maxwellian distribution, we must start by assuming the system is at equilibrium. By detail balance, we get:

$$A_{jk} n_j n_k = A_{k-1,j+1} n_{k-1} n_{j+1} = B_{j+1,k-1} n_{k-1} n_{j+1} \qquad \text{(B-8)}$$

By using equations (B-3) and (B-6) and rearranging, this eventually becomes:

$$A_{jk} = A_{k-1,j+1} \left(\frac{\varepsilon_{k-1}}{\varepsilon_k} \right)^{1/2} \left(\frac{\varepsilon_{j+1}}{\varepsilon_j} \right)^{1/2} \qquad \text{(B-9)}$$

This equation is important in modeling electron-electron interactions, since it means that only half of the interaction matrix is to be calculated using previously-derived formulas. In actual practice, the upper-triangular half of the A_{kj} matrix is calculated using equation (B-3), and the rest

calculated by using equation (B-9). If equation (B-3) alone is used to calculate the interaction matrix, the distribution function will not converge properly at lowest and highest energies.

APPENDIX C: A User's Guide to MEGABOLTZ, or How to Simulate a Plasma in the Safety of Your Own Home

Files Needed

To run this version of MEGABOLTZ, you need a mainframe running UNIX version 4.3 or above, and the following files:

FILE	DESCRIPTION
megaboltz	the program
input.com	input datafile read by megaboltz
*.crs	one file per gas, containing cross-sectional data

and finally, an image-translation program such as mit or mps to convert the plot file into something usable by a laser printer.

As a general percaution, keep an extra copy of the input file around, in case you torch the original. Also, should you wish to include your own library files for use by MEGABOLTZ, Appendix D gives a brief description on how these files should be formatted.

A Sample Run

Let's say we want to calculate electron-transport parameters through a mixture of 80% Nitrogen and 20% Oxygen at a pressure of 760 Torr and temperature of 300 K. Let E/N equal 10 Townsends (10^{-16} V-cm^2), assume an electron number density of 10^{-14} cm^3, and a Druyvesteyn distribution of electrons for your initial guess. Solve for 100 bins of width 0.1 eV, and assume a time step of 10^{-6} seconds.

First, open the file 'input.com' using your favorite text editor. You should then see something like the following (Figure 17):

```
*--- Input parameters
     E/N ratio (V-cm^2): 1.d-17
    Gas Temperature (K): 0.
    Gas Pressure (torr): 1.d0
 e- num density (cm^-3): 1.d14
   Vibrational Temp (K): 0.

*--- Mesh parameters
         Number of bins: 100
         Bin Width (eV): .2
          Time step (s): 1.d-3

*--- Program switches
  e-e collisions? (Y/N): N
 Initial e- dist. (M/D): M
          Save graph as: test3.plot
           Save data as: test3.data


*--- Gasmix data by type (5 maximum), percent (enter as XXX), and
whether
      you want to view its cross-sections (enter either Y or N)
_Gas_   _Percent_ _View?_
n2          100         N
EOF           0         N
```

Figure 17: Input File for MEGABOLTZ before Editing.

Edit the appropriate lines for E/N thru time step by going to the end of the line, deleting the previous value, and putting the new value in. MEGABOLTZ will look at the entire line, mask out the first 25 characters (which describe the variable being input), and put the rest into the appropriate program variable. The first 25 characters of these lines, as a result, should not change when you edit the file. For initial guess, choose 'D' (in practice, you COULD type in the word 'Druyvesteyn', since the program will only read the first character after the mask as data on that line). Choose file names you will remember for the plot file and save file. For the gas mix table, follow the positioning of the data already in the table. For purposes of this walk-through, you do not want to look at the interpolated cross-sections. When you're through with all the edits, the input file should look like Figure 18. If this is so, save and quit the text editor. If not, go back and keep editing until it does.

```
*--- Input parameters
     E/N ratio (V-cm^2): 1.d-16
    Gas Temperature (K): 3.d2
    Gas Pressure (torr): 7.6d2
 e- num density (cm^-3): 1.d14
   Vibrational Temp (K): 0.

*--- Mesh parameters
         Number of bins: 200
         Bin Width (eV): .1
          Time step (s): 1.d-10

*--- Program switches
  e-e collisions? (Y/N): Y
 Initial e- dist. (M/D): D
          Save graph as: test3.plot
           Save data as: test3.data


*--- Gasmix data by type (5 maximum), percent (enter as XXX), and
whether
     you want to view its cross-sections (enter either Y or N)
_Gas_   _Percent_ _View?_
n2            80         N
o2            20         N
EOF            0         N
```

Figure 18: Input File for MEGABOLTZ after Editing.

```
galaxy-43 megaboltz
  Energy axis initialized
  Cross-sections loaded and interpolated
  Initial guess calculated
  Elastic rates loaded
  Inelastic rates loaded
  Superelastic rates loaded
  Ionization rates loaded
  [I - hC] calculated
  e-e collision matrices calculated
  [I - hC] decomposed, starting time iteration
  Time iteration complete
  Energy balance calculated
  Written to save file  test3.data
  Generating plots as  test3.plot
```

Figure 19: What a Typical MEGABOLTZ Run Looks Like

Now it's time to run the code. At your system prompt, type 'megaboltz' (small letters only -- UNIX cares...) and press RETURN. As the program runs, your screen will display the status lines shown in figure 19, letting you know where you are in the run. When it ends, you can review the

save file with UNIX utilities like more or less, call the file up in a text editor, or download the file to your PC at work or home. The textfile generated by MEGABOLTZ is tab-delimited, allowing it to be read into most PC spreadsheets for further analysis.

To print the plot file, invoke one of the following sequences of UNIX commands:

mit GRAPHFILE > lpr -Pimagen	to the IMAGEN printer in Bldg 642, room 2202. Please note this command is specific to AFIT.
mps GRAPHFILE > lpr -Plw	to the PostScript printer named 'lw'. Should work with any PostScript printer, provided you know its name

An example of the generated plot file is included in Figure 20. The textfile output from this example is included in its entirety in the next section.

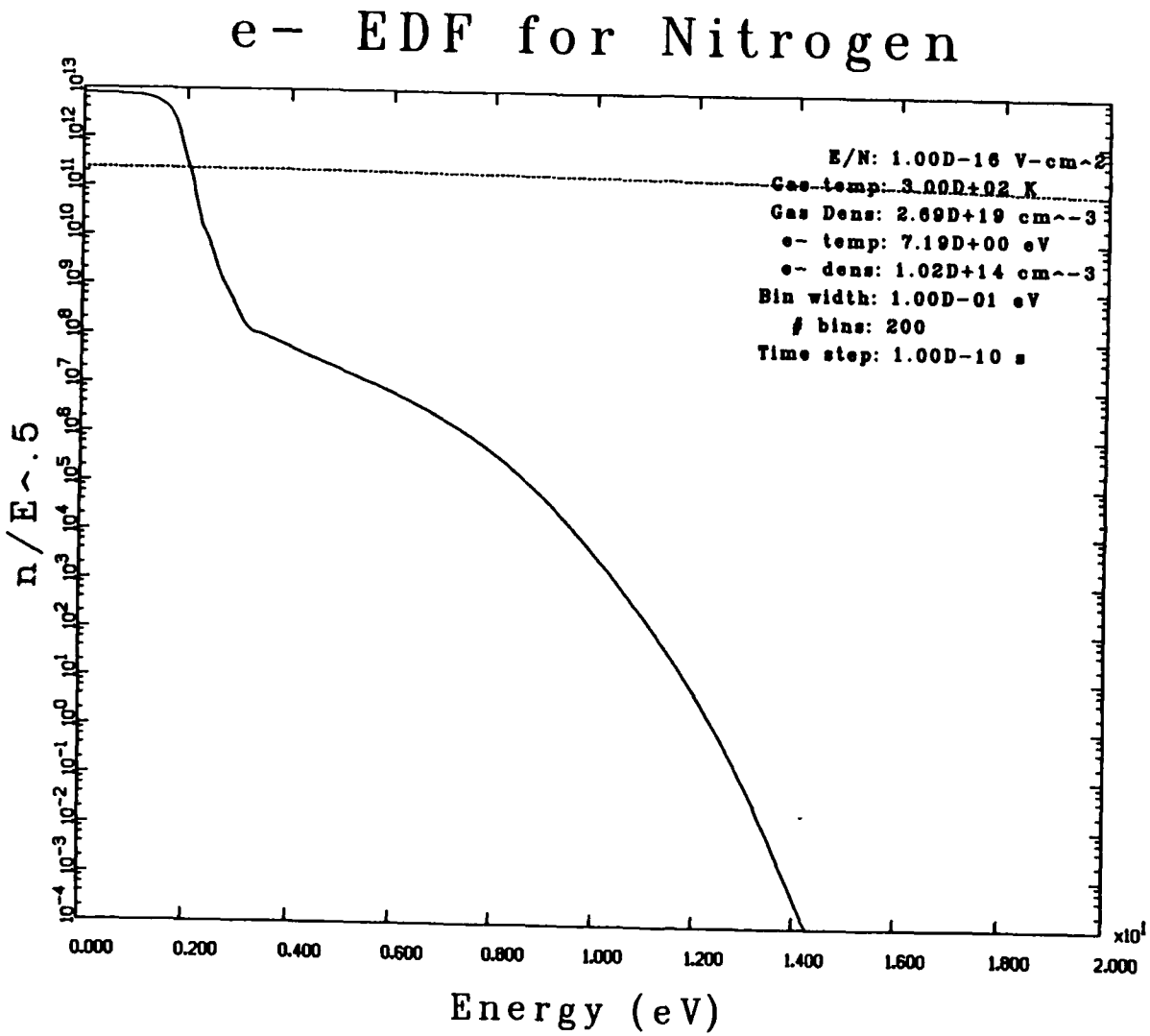


Figure 20: Plot File Generated by MEGABOLTZ

Sample Output

```
            E/N          Temp            N           ne
      1.0000D-16  3.0000D+02  2.6868D+19  1.0223D+14

      Bins    Binwidth    Timestep
      200    1.00D-01    1.00D-10

  e- temp:           7.1932619708381         ev

# iterations:      37   Successful convergence
  All distributions are reduced
        EN[i]        final    analytic     Rel Err
 1   1.00D-01     7.8820D+12     2.3154D+11     2.3071D+01
 2   2.00D-01     7.8592D+12     2.3152D+11     2.8397D+01
 3   3.00D-01     7.8335D+12     2.3150D+11     2.9890D+01
 4   4.00D-01     7.8021D+12     2.3147D+11     3.0530D+01
 5   5.00D-01     7.7676D+12     2.3142D+11     3.0842D+01
 6   6.00D-01     7.7300D+12     2.3137D+11     3.0987D+01
 7   7.00D-01     7.6874D+12     2.3131D+11     3.1025D+01
 8   8.00D-01     7.6212D+12     2.3124D+11     3.0912D+01
 9   9.00D-01     7.5395D+12     2.3116D+11     3.0697D+01
10   1.00D+00     7.4266D+12     2.3107D+11     3.0327D+01
11   1.10D+00     7.2791D+12     2.3097D+11     2.9791D+01
12   1.20D+00     7.0213D+12     2.3086D+11     2.8774D+01
13   1.30D+00     6.6124D+12     2.3074D+11     2.7101D+01
14   1.40D+00     6.0347D+12     2.3061D+11     2.4697D+01
15   1.50D+00     5.3222D+12     2.3047D+11     2.1705D+01
16   1.60D+00     4.4648D+12     2.3032D+11     1.8080D+01
17   1.70D+00     3.4757D+12     2.3016D+11     1.3877D+01
18   1.80D+00     2.2478D+12     2.3000D+11     8.6366D+00
19   1.90D+00     9.6019D+11     2.2982D+11     3.1226D+00
20   2.00D+00     3.5286D+11     2.2964D+11     5.1725D-01
21   2.10D+00     1.6054D+11     2.2944D+11     3.0867D-01
22   2.20D+00     5.0429D+10     2.2924D+11     7.8253D-01
23   2.30D+00     1.5826D+10     2.2903D+11     9.3166D-01
24   2.40D+00     9.2473D+09     2.2880D+11     9.6001D-01
25   2.50D+00     4.8106D+09     2.2857D+11     9.7917D-01
26   2.60D+00     2.2834D+09     2.2833D+11     9.9010D-01
27   2.70D+00     1.1736D+09     2.2808D+11     9.9490D-01
28   2.80D+00     7.3238D+08     2.2782D+11     9.9681D-01
29   2.90D+00     4.8187D+08     2.2755D+11     9.9790D-01
30   3.00D+00     2.8857D+08     2.2728D+11     9.9874D-01
31   3.10D+00     1.7265D+08     2.2699D+11     9.9925D-01
32   3.20D+00     1.2458D+08     2.2669D+11     9.9945D-01
33   3.30D+00     1.0750D+08     2.2639D+11     9.9953D-01
34   3.40D+00     1.0198D+08     2.2608D+11     9.9955D-01
35   3.50D+00     9.2547D+07     2.2576D+11     9.9959D-01
36   3.60D+00     8.3077D+07     2.2542D+11     9.9963D-01
37   3.70D+00     7.4594D+07     2.2508D+11     9.9967D-01
38   3.80D+00     6.6924D+07     2.2474D+11     9.9970D-01
39   3.90D+00     6.0044D+07     2.2438D+11     9.9973D-01
40   4.00D+00     5.3981D+07     2.2401D+11     9.9976D-01
41   4.10D+00     4.8665D+07     2.2364D+11     9.9978D-01
42   4.20D+00     4.3901D+07     2.2325D+11     9.9980D-01
43   4.30D+00     3.9637D+07     2.2286D+11     9.9982D-01
```

44	4.40D+00	3.5819D+07	2.2246D+11	9.9984D-01
45	4.50D+00	3.2400D+07	2.2205D+11	9.9985D-01
46	4.60D+00	2.9336D+07	2.2164D+11	9.9987D-01
47	4.70D+00	2.6558D+07	2.2121D+11	9.9988D-01
48	4.80D+00	2.4039D+07	2.2078D+11	9.9989D-01
49	4.90D+00	2.1754D+07	2.2033D+11	9.9990D-01
50	5.00D+00	1.9681D+07	2.1988D+11	9.9991D-01
51	5.10D+00	1.7801D+07	2.1943D+11	9.9992D-01
52	5.20D+00	1.6094D+07	2.1896D+11	9.9993D-01
53	5.30D+00	1.4544D+07	2.1848D+11	9.9993D-01
54	5.40D+00	1.3137D+07	2.1800D+11	9.9994D-01
55	5.50D+00	1.1859D+07	2.1751D+11	9.9995D-01
56	5.60D+00	1.0697D+07	2.1701D+11	9.9995D-01
57	5.70D+00	9.6418D+06	2.1651D+11	9.9996D-01
58	5.80D+00	8.6824D+06	2.1599D+11	9.9996D-01
59	5.90D+00	7.8099D+06	2.1547D+11	9.9996D-01
60	6.00D+00	7.0164D+06	2.1494D+11	9.9997D-01
61	6.10D+00	6.2946D+06	2.1440D+11	9.9997D-01
62	6.20D+00	5.6342D+06	2.1386D+11	9.9997D-01
63	6.30D+00	5.0305D+06	2.1331D+11	9.9998D-01
64	6.40D+00	4.4803D+06	2.1275D+11	9.9998D-01
65	6.50D+00	3.9801D+06	2.1218D+11	9.9998D-01
66	6.60D+00	3.5265D+06	2.1161D+11	9.9998D-01
67	6.70D+00	3.1162D+06	2.1103D+11	9.9999D-01
68	6.80D+00	2.7458D+06	2.1044D+11	9.9999D-01
69	6.90D+00	2.4123D+06	2.0985D+11	9.9999D-01
70	7.00D+00	2.1125D+06	2.0924D+11	9.9999D-01
71	7.10D+00	1.8436D+06	2.0863D+11	9.9999D-01
72	7.20D+00	1.6027D+06	2.0802D+11	9.9999D-01
73	7.30D+00	1.3871D+06	2.0740D+11	9.9999D-01
74	7.40D+00	1.1946D+06	2.0677D+11	9.9999D-01
75	7.50D+00	1.0232D+06	2.0613D+11	1.0000D+00
76	7.60D+00	8.7182D+05	2.0549D+11	1.0000D+00
77	7.70D+00	7.3906D+05	2.0484D+11	1.0000D+00
78	7.80D+00	6.2340D+05	2.0419D+11	1.0000D+00
79	7.90D+00	5.2324D+05	2.0352D+11	1.0000D+00
80	8.00D+00	4.3701D+05	2.0286D+11	1.0000D+00
81	8.10D+00	3.6317D+05	2.0218D+11	1.0000D+00
82	8.20D+00	3.0029D+05	2.0150D+11	1.0000D+00
83	8.30D+00	2.4703D+05	2.0082D+11	1.0000D+00
84	8.40D+00	2.0217D+05	2.0012D+11	1.0000D+00
85	8.50D+00	1.6457D+05	1.9943D+11	1.0000D+00
86	8.60D+00	1.3324D+05	1.9872D+11	1.0000D+00
87	8.70D+00	1.0729D+05	1.9801D+11	1.0000D+00
88	8.80D+00	8.5928D+04	1.9730D+11	1.0000D+00
89	8.90D+00	6.8452D+04	1.9658D+11	1.0000D+00
90	9.00D+00	5.4240D+04	1.9585D+11	1.0000D+00
91	9.10D+00	4.2753D+04	1.9512D+11	1.0000D+00
92	9.20D+00	3.3527D+04	1.9439D+11	1.0000D+00
93	9.30D+00	2.6161D+04	1.9364D+11	1.0000D+00
94	9.40D+00	2.0315D+04	1.9290D+11	1.0000D+00
95	9.50D+00	1.5701D+04	1.9214D+11	1.0000D+00
96	9.60D+00	1.2081D+04	1.9139D+11	1.0000D+00
97	9.70D+00	9.2544D+03	1.9063D+11	1.0000D+00
98	9.80D+00	7.0597D+03	1.8986D+11	1.0000D+00
99	9.90D+00	5.3644D+03	1.8909D+11	1.0000D+00
100	1.00D+01	4.0615D+03	1.8831D+11	1.0000D+00

101	1.01D+01	3.0657D+03	1.8753D+11	1.0000D+00
102	1.02D+01	2.3041D+03	1.8675D+11	1.0000D+00
103	1.03D+01	1.7243D+03	1.8596D+11	1.0000D+00
104	1.04D+01	1.2849D+03	1.8516D+11	1.0000D+00
105	1.05D+01	9.5343D+02	1.8436D+11	1.0000D+00
106	1.06D+01	7.0448D+02	1.8356D+11	1.0000D+00
107	1.07D+01	5.1832D+02	1.8276D+11	1.0000D+00
108	1.08D+01	3.7970D+02	1.8195D+11	1.0000D+00
109	1.09D+01	2.7690D+02	1.8113D+11	1.0000D+00
110	1.10D+01	2.0096D+02	1.8031D+11	1.0000D+00
111	1.11D+01	1.4505D+02	1.7949D+11	1.0000D+00
112	1.12D+01	1.0409D+02	1.7867D+11	1.0000D+00
113	1.13D+01	7.4269D+01	1.7784D+11	1.0000D+00
114	1.14D+01	5.2682D+01	1.7700D+11	1.0000D+00
115	1.15D+01	3.7144D+01	1.7617D+11	1.0000D+00
116	1.16D+01	2.6018D+01	1.7533D+11	1.0000D+00
117	1.17D+01	1.8092D+01	1.7449D+11	1.0000D+00
118	1.18D+01	1.2470D+01	1.7364D+11	1.0000D+00
119	1.19D+01	8.4899D+00	1.7279D+11	1.0000D+00
120	1.20D+01	5.6926D+00	1.7194D+11	1.0000D+00
121	1.21D+01	3.8024D+00	1.7109D+11	1.0000D+00
122	1.22D+01	2.5297D+00	1.7023D+11	1.0000D+00
123	1.23D+01	1.6757D+00	1.6937D+11	1.0000D+00
124	1.24D+01	1.1046D+00	1.6851D+11	1.0000D+00
125	1.25D+01	7.2387D-01	1.6764D+11	1.0000D+00
126	1.26D+01	4.7033D-01	1.6678D+11	1.0000D+00
127	1.27D+01	3.0303D-01	1.6591D+11	1.0000D+00
128	1.28D+01	1.9358D-01	1.6503D+11	1.0000D+00
129	1.29D+01	1.2257D-01	1.6416D+11	1.0000D+00
130	1.30D+01	7.6857D-02	1.6328D+11	1.0000D+00
131	1.31D+01	4.7602D-02	1.6241D+11	1.0000D+00
132	1.32D+01	2.9131D-02	1.6153D+11	1.0000D+00
133	1.33D+01	1.7623D-02	1.6064D+11	1.0000D+00
134	1.34D+01	1.0545D-02	1.5976D+11	1.0000D+00
135	1.35D+01	6.2484D-03	1.5887D+11	1.0000D+00
136	1.36D+01	3.6785D-03	1.5799D+11	1.0000D+00
137	1.37D+01	2.1522D-03	1.5710D+11	1.0000D+00
138	1.38D+01	1.2525D-03	1.5621D+11	1.0000D+00
139	1.39D+01	7.2669D-04	1.5532D+11	1.0000D+00
140	1.40D+01	4.2058D-04	1.5442D+11	1.0000D+00
141	1.41D+01	2.4318D-04	1.5353D+11	1.0000D+00
142	1.42D+01	1.4051D-04	1.5263D+11	1.0000D+00
143	1.43D+01	8.1195D-05	1.5174D+11	1.0000D+00
144	1.44D+01	4.6950D-05	1.5084D+11	1.0000D+00
145	1.45D+01	2.7209D-05	1.4994D+11	1.0000D+00
146	1.46D+01	1.5872D-05	1.4904D+11	1.0000D+00
147	1.47D+01	9.3203D-06	1.4814D+11	1.0000D+00
148	1.48D+01	5.5086D-06	1.4724D+11	1.0000D+00
149	1.49D+01	3.2754D-06	1.4634D+11	1.0000D+00
150	1.50D+01	1.9565D-06	1.4544D+11	1.0000D+00
151	1.51D+01	1.1692D-06	1.4454D+11	1.0000D+00
152	1.52D+01	6.9941D-07	1.4363D+11	1.0000D+00
153	1.53D+01	4.1879D-07	1.4273D+11	1.0000D+00
154	1.54D+01	2.5100D-07	1.4183D+11	1.0000D+00
155	1.55D+01	1.5056D-07	1.4093D+11	1.0000D+00
156	1.56D+01	9.0351D-08	1.4002D+11	1.0000D+00
157	1.57D+01	5.4220D-08	1.3912D+11	1.0000D+00

158	1.58D+01	3.2538D-08	1.3822D+11	1.0000D+00
159	1.59D+01	1.9524D-08	1.3732D+11	1.0000D+00
160	1.60D+01	1.1713D-08	1.3641D+11	1.0000D+00
161	1.61D+01	7.0234D-09	1.3551D+11	1.0000D+00
162	1.62D+01	4.2089D-09	1.3461D+11	1.0000D+00
163	1.63D+01	2.5209D-09	1.3371D+11	1.0000D+00
164	1.64D+01	1.5090D-09	1.3281D+11	1.0000D+00
165	1.65D+01	9.0280D-10	1.3191D+11	1.0000D+00
166	1.66D+01	5.3982D-10	1.3101D+11	1.0000D+00
167	1.67D+01	3.2260D-10	1.3011D+11	1.0000D+00
168	1.68D+01	1.9267D-10	1.2921D+11	1.0000D+00
169	1.69D+01	1.1499D-10	1.2832D+11	1.0000D+00
170	1.70D+01	6.8566D-11	1.2742D+11	1.0000D+00
171	1.71D+01	4.0825D-11	1.2652D+11	1.0000D+00
172	1.72D+01	2.4272D-11	1.2563D+11	1.0000D+00
173	1.73D+01	1.4409D-11	1.2474D+11	1.0000D+00
174	1.74D+01	8.5414D-12	1.2385D+11	1.0000D+00
175	1.75D+01	5.0555D-12	1.2296D+11	1.0000D+00
176	1.76D+01	2.9873D-12	1.2207D+11	1.0000D+00
177	1.77D+01	1.7624D-12	1.2118D+11	1.0000D+00
178	1.78D+01	1.0380D-12	1.2030D+11	1.0000D+00
179	1.79D+01	6.1034D-13	1.1941D+11	1.0000D+00
180	1.80D+01	3.5824D-13	1.1853D+11	1.0000D+00
181	1.81D+01	2.0983D-13	1.1765D+11	1.0000D+00
182	1.82D+01	1.2266D-13	1.1677D+11	1.0000D+00
183	1.83D+01	7.1549D-14	1.1589D+11	1.0000D+00
184	1.84D+01	4.1651D-14	1.1501D+11	1.0000D+00
185	1.85D+01	2.4197D-14	1.1414D+11	1.0000D+00
186	1.86D+01	1.4028D-14	1.1327D+11	1.0000D+00
187	1.87D+01	8.1161D-15	1.1240D+11	1.0000D+00
188	1.88D+01	4.6857D-15	1.1153D+11	1.0000D+00
189	1.89D+01	2.6994D-15	1.1067D+11	1.0000D+00
190	1.90D+01	1.5517D-15	1.0980D+11	1.0000D+00
191	1.91D+01	8.8997D-16	1.0894D+11	1.0000D+00
192	1.92D+01	5.0929D-16	1.0808D+11	1.0000D+00
193	1.93D+01	2.9083D-16	1.0723D+11	1.0000D+00
194	1.94D+01	1.6576D-16	1.0637D+11	1.0000D+00
195	1.95D+01	9.4369D-17	1.0552D+11	1.0000D+00
196	1.96D+01	5.3769D-17	1.0467D+11	1.0000D+00
197	1.97D+01	3.0830D-17	1.0382D+11	1.0000D+00
198	1.98D+01	1.8058D-17	1.0298D+11	1.0000D+00
199	1.99D+01	1.1208D-17	1.0214D+11	1.0000D+00
200	2.00D+01	8.2129D-18	1.0130D+11	1.0000D+00

E-avg of Numerical := 1.0304D+00 eV
Drift velocity := 2.1358D+06 cm/s
Diffusion coeff := 1.1828D+02 cm^2/s

***** ENERGY BALANCE *****
Energy gain := 5.7387D+09 eV/s
Elastic losses := 6.0033D+07 eV/s
Inelastic losses := 6.6059D+09 eV/s
Energy balance := -9.2729D+08

Timed Jobs

Occasionally, you might need to run a case dominated by electron-electron interactions on a slow computer system, such as BLACKBIRD. Instead of waiting 90 minutes to three hours for your output and wasting time hovering over a terminal in the process, UNIX allows you to set a timed job, which the computer will run for you at a time you specify. Documentation for the 'at' command is provided on your mainframe, but I will briefly go through the steps needed to set a timed job.

At the system prompt, type the following:

%at -c -m xxxx

where xxxx is the time you want your job to run, in 24-hour format. You will then get a prompt that looks like this: 'at>'. At this prompt, type 'megaboltz' and press RETURN. You will get another 'at>' prompt. At this time, you can (if you want) type in the command for printing the plot file. Otherwise, if you are done typing in commands you want the computer to execute at the given time, type CONTROL-D. You will be returned to the system prompt '%' (or whatever you use) at that time. Make sure the input file contains the data with which you want to calculate this particular EDF, make sure the output file names won't be duplicated, and you're free to do other things. When the program finishes executing, the computer will mail you, telling you whether your command sequence successfully ran or not. If not, the letter will include the error message generated by the program upon termination.

Error Messages

While many of us would like to believe that this is a perfect world, experience has often shown us otherwise. Running the MEGABOLTZ code is no exception to this rule. To make living in an imperfect world easier, MEGABOLTZ does some of its own error trapping. A listing of these error conditions, what happens when they occur, and possible causes are listed below:

***** ERROR reading input data *****

Something has probably been misplaced in the 'input.com' file, probably caused by the program sensing a carriage return in the middle of a specifically-formatted field. Program will terminate on this error. Check the input file for badly-formatted lines, using Figures 17 and 18 above as your guide.

******* ERROR opening input datafile *******

The file 'input.com' is either missing or damaged. Program will terminate on this error. If you kept a copy around somewhere, you're OK.

******* ERROR opening cross-section file *******

File: -filename-

The library file -filename- is either missing or damaged. Program will terminate on this error. Find a working duplicate of the needed file (you DID make some, didn't you?).

******* ERROR reading cross-section data *******

File: -filename-

Data in the header of file -filename- is not in correct format. Program will terminate on this error. See Appendix D for description of correct file format.

******* ERROR reading cross-section data *******

File: -filename-, Interaction: -j-

Data in table -j- of file -filename- is not reading in correctly. Program will terminate on this error. Check how you have the data formatted, and check the header of the table preceding it. Sometimes, the number of data pairs for the preceding table may be off, throwing off successive data reads.

******* E R R O R ! *******

Negative number detected in e- distribution

Iteration # -h-

Solution may be numerically unstable

A negative number has been detected in the calculated EDF at iteration number -h-. Program terminates on this error. A negative electron number density at any energy is an unphysical result, and often results from one or more of the following conditions: excessively wide bin width, excessively large time step (especially in cases where electron-electron interactions play a signifigant role in calculating the distribution), and/or an excessively high value for E/N.

******* Filename -filename- exists! *******

Please type new filename:

The graph or save file in question already exists on the computer. Rather than overwrite the old file, the program prompts the user for a different filename. Once the user inputs the new filename, MEGABOLTZ loops back to the point where this error originally occurred and continues execution.

******* ERROR writing to save file *******

Attempting to salvage what I can...

For some reason, there has been a problem writing data to the save file. MEGABOLTZ immediately closes the save file and terminates, on the assumption that whatever data is already in the file is better than none at all. How much is saved depends on when the error occurred during writing.

***** Arithmetic Exception:**

Floating-point overflows, underflows, and divisions by zero will cause this error to occur. MEGABOLTZ will terminate upon this error happening. The error message will describe one of the above conditions as reason for its appearing. Check the input values in your input file, and if the problem keeps occurring, check to make sure your cross-sections are being read in properly. Anything else would require tearing into the source code for MEGABOLTZ.

***** MIT:**

Any error which occurs during generation of the plot file will start with this flag, then get more specific from there. Said types of error will also cause MEGABOLTZ to terminate execution.

This is a moderately exhaustive list of errors encountered during development and validation of MEGABOLTZ, and which might reasonably be expected to occur should something go wrong. If an error occurs which is not covered in the above list, note exactly what the computer says upon its occurrence, then contact your computer's sysop.

APPENDIX D: Cross-section Library Format

Should anyone need to write their own cross-sectional library file for MEGABOLTZ's use, this appendix goes over what should be included in the file, and how it should be formatted. MEGABOLTZ unfortunately follows rather strict instructions on reading data from a library file, which make it easy for the unwary to make a mistake. Library files for MEGABOLTZ are simple text files, and can be viewed and edited as such. To aid such an endeavor, looking at the library files that come with MEGABOLTZ is not only instructive, but encouraged.

A quick explanation of the terms used for formatting: 'A12' or 'A15' means data should be in a twelve- or fifteen-character string respectively; 'iii' means an integer number up to three characters in length, and 'do.d' means a real or double-precision number with two characters before the decimal point and one after. If blanks are required in a line, they are explicitly called for in that line's description. Terminate all library file 'lines' with a blank followed by the '/' character.

Line 1: Chemical name of the gas. (eg. N2 or CO2)

Line 2: Molecular Weight in AMU (format dd.d)

Number of interaction tables in the file (format iii)

Ionization Energy (free format - put at least one blank in to separate from the previous number on the line)

EXAMPLE: Nitrogen gas has a molecular weight of 28 AMU, 25 interaction tables for MEGABOLTZ's use, and an ionization energy of 18.75 eV. Following the above format, the second line in a Nitrogen library file would look like:

28.^^25^18.75^/ (continued...)

(continued from previous page...)

where the '^' character, for purposes of this example, represents a blank. Yes, I know what I said about blanks above, so perhaps it might help those lawyers who pointed that out by visualizing the input line as follows:

28.0025018.75^/

Whether they be blanks or zeros, some padding is necessary to insure numbers end up in the variables they are supposed to...

Line 3: Number of data pairs in the table (format iii)

Description of process (inelastic, ionization, attachment, etc.; format A12)

Process name (eg. 'N2 v(v=1)', 'N2 A3SIGMA'; format A15)

Source of data (eg. 'KIEFFER 1973'; format A15)

Scale factor (usually 1.0, but can sometimes vary; format dd.d)

EXAMPLE: Line 3, for the rotational inelastic interaction table of Nitrogen, would look like:

53^inelastic^^^N2 ROT^^^^^^^BAILEY^^^^^^^1.0^/

where the '^' character once again represents blanks. Again, the extra blanks are necessary to pad the data out to fit the format MEGABOLTZ expects.

Line 4: Data in generalized form ENERGY, CROSS-SECTION. Energy is in units of eV, while cross-section is in units of 1.e-16 cm2. First pair must have energy of 0 eV, and each data pair must be separated by at least one blank. Line 4 will take up

several lines of the library file, so don't be afraid to use carriage returns to maintain some semblance of readability. Just this once, MEGABOLTZ doesn't care that much about how the data pairs appear, as long as they're two numbers separated by a comma and there is at least one space between the pairs.

EXAMPLE: Part of the data table for Nitrogen momentum transfer collisions might appear as:

0.,1.^^.001,1.1^^.01,1.2^^... (etc, etc)

where once again, the '^' character represents a blank.

Line 5: Energy loss of the process in eV (format dd.d). IMPORTANT NOTE: The first cross-section table listed in a gas file MUST be momentum transfer, and it WILL NOT have this line.

Repeat lines 3 - 5 for all interactions of that gas. Terminate the file with the string, "--EOF--" on its own line.

Bibliography

1. Allis, W. P., "Motion of Ions and Electrons", Handbuch Der Physik Vol XXI, Springer-Velag (Berlin), 1956.

2. Bretagne, J. *et al* , "Low-Energy Electron Distributions in an Ar Plasma", Journal of Physics D: Applied Physics, 15: 2205 - 2225 (1982).

3. --------, "Electron Energy Distributions in Electron Beam Sustained Discharges", Journal of Physics D: Applied Physics, 18: 811 - 825 (1985).

4. Davies, A., Smythe, K., and Thompson, R., "BOLTZ: A Code to Solve the Transport Equation for Electron Distributions and Then Calculate Transport Coefficients and Vibrational Excitation Rates in Gases with Applied Fields", Computer Physics Communications, 11: 369 - 383 (1976).

5. Druyvesteyn, M. J., Physica 10:69 (1930)

6. Elliott, C. J., and Greene, A. E., "Electron Energy Distribution Functions in e-Beam Generated Xe and Ar Plasmas", Journal of Applied Physics, 47: 2946 - 2953 (July 1976).

7. Huxley, L. G. H., and Crompton, E. W., The Diffusion and Drift of Electrons in Gases, Wiley (New York), 1973.

8. Holstein, T. "Energy Distribution of Electrons in High-Frequency Gas Discharges", Physical Review, 70: 367 - 384 (1946).

9. Long, W., Bailey, W., and Garscadden, A., "Electron Drift Velocities in Molecular Gas - Rare Gas Mixtures", Physical Review A, 13: 471 - 475 (1975).

10. Morgan, W. L., JILA Information Center Report # 19, June 1979.

11. Nickel, G., "Transport of Spectral Line Radiation in Cylindrical Plasmas", Report No. LA-UR-86-580, Los Alamos National Laboratory, 1986.

12. Press, W. *et al* , Numerical Recipes, Cambridge University Press, 1986.

13. Reif, F., Fundamentals of Statistical and Thermal Physics, McGraw-Hill, 1965.

14. Rockwood, S. D., "Elastic and Inelastic Cross Sections for Electron-Hg Scattering from Hg Transport Data", Physical Review A, 8: 2348 - 2358 (1973).

15. Rockwood, S. D., and Greene, A. E., "Numerical Solutions of the Boltzmann Transport Equation", Computer Physics Communications, 19: 373 - 390 (1980).

16. Smirnov, B. M., Physics of Weakly Ionized Gases, Mir Publishers (Moscow), 1981.

Vita

Captain Gaylord E. Seger, III, [redacted] attended Purdue University on an Air Force ROTC scholarship. He was awarded the degree of Bachelor of Science in Physics in May 1984. Upon graduation, he received a commission in the USAF through the ROTC program, and was immediately called to extended active duty. He served as Test Scheduling Officer and cost/schedule analyst at the AFWL Electromagnetic Pulse Testing Facilities, Air Force Weapons Lab, Kirtland AFB, New Mexico from 1984 to 1987. In 1987, he was transferred within AFWL to the Technology Assessment Office, where he served as Business Manager until entering the School of Engineering, Air Force Institute of Technology, in May 1988.

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GEP/ENP/89D-10	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering 6b. OFFICE SYMBOL (*If applicable*) AFIT/ENP	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (*City, State, and ZIP Code*)	7b. ADDRESS (*City, State, and ZIP Code*)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (*If applicable*)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (*City, State, and ZIP Code*)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (*Include Security Classification*)
A NUMERICAL SOLUTION TO THE TIME-DEPENDENT BOLTZMANN EQUATION

12. PERSONAL AUTHOR(S)
GAYLORD E. SEGER, III, B.S., Capt, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (*Year, Month, Day*)	15. PAGE COUNT
MS Thesis	FROM _____ TO _____	1989 December	87

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUB-GROUP	
12	01		Numerical Methods and Procedures, Boltzmann Equation, Plasma Physics, Transport Properties, Electric Discharges
20	09		

19. ABSTRACT (*Continue on reverse if necessary and identify by block number*)

Thesis Advisor: William F. Bailey
Associate Professor of Physics
Department of Engineering Physics

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL William F. Bailey, Assoc. Professor	22b TELEPHONE (*Include Area Code*) (513)255-2012 / 22c OFFICE SYMBOL ENP

DD Form 1473, JUN 86 *Previous editions are obsolete.*

Interest in gas discharge phenomena for laser, ion source, and plasma processing applications has generated needs for the solution of the time-dependent Boltzmann Equation. An algorithm is developed and tested to compute the electron energy distribution function (EDF), incorporating elastic, inelastic, superelastic, ionization, and electron-electron collisions. A new finite-differencing approach which eliminates low-energy instabilities inherent in previous techniques is developed and tested. An implicit-explicit Euler approximation technique was used for the algorithm to transform the resulting nonlinear differential equation into a system of finite-differenced linear equations. The system is then solved using LU-decomposition for efficient matrix computation. The program developed using this algorithm, MEGABOLTZ, is first put through basic shakedown tests to verify the correctness of the algorithm. Next, the program calculates an EDF for nitrogen gas for electric field to neutral number density ratios (E/N) ranging from 5-40 Townsend. Distributions computed by MEGABOLTZ were compared to previously-reported data from other Boltzmann Equation solvers and experiments, and are found to be in good agreement with them. Future modifications are suggested for MEGABOLTZ to improve robustness and accuracy. A short user's manual for MEGABOLTZ is also included.

Keywords: plasma physics, electric discharges, transport properties, (KR)